ELSEVIER

Operations Research Letters 24 (1999) 37–45

operations research letters

Dual decomposition in stochastic integer programming [1]

Claus C. Carøe[a], Rüdiger Schultz[b, *]

[a] University of Copenhagen, Institute of Mathematics, Universitetsparken 5, DK-2100 Copenhagen, Denmark
[b] Department of Mathematics, Gerhard-Mercator University Duisburg, Lotharstr. 65, D-47048 Duisburg, Germany

Received 1 December 1996; received in revised form 1 September 1998

Abstract

We present an algorithm for solving stochastic integer programming problems with recourse, based on a dual decomposition scheme and Lagrangian relaxation. The approach can be applied to multi-stage problems with mixed-integer variables in each time stage. Numerical experience is presented for some two-stage test problems. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Stochastic programming; Mixed-integer programming; Lagrangian relaxation; Branch and bound

1. Introduction

Stochastic programs with recourse are aimed at finding non-anticipative here-and-now decisions that must be taken prior to knowing the realizations of some random variables such that total expected costs (revenues) from here-and-now decisions and possible *recourse* actions are minimized (maximized). When some of the decision variables are required to be integer or binary we speak of a stochastic integer programming problem [3, 17].

Stochastic integer programs are challenging from both computational and theoretical points of view since they combine two difficult type of models into one. Until now algorithmic results have been limited to special instances. Laporte and Louveaux [11] developed an integer L-shaped decomposition algorithm for problems with binary first stage and easily computable recourse costs. Løkketangen and Woodruff [12] applied the progressive hedging algorithm and tabu search to multi-stage problems with mixed 0–1 variables. Takriti et al. [18] report about application of progressive hedging to multi-stage stochastic unit commitment problems in power generation.

A framework for solving two-stage problems using Gröbner bases was proposed by Schultz et al. [16], but is limited to problems with integer recourse variables only. For problems with simple integer recourse the expected recourse costs can be made separable and special methods like the one in Klein Haneveld et al. [10] can be applied.

In the present paper we develop a decomposition method that applies to general linear two- and multi-stage stochastic programs with integrality requirements. Decomposition methods for stochastic programs generally fall in two groups: *Primal*

[1] This research was carried out while the authors were at Konrad-Zuse-Zentrum für Informationstechnik Berlin.

* Corresponding author. E-mail: schultz@math.uni-duisburg.de.

PII: S0167-6377(98)00050-9

methods that work with subproblems assigned to time stages and *dual* methods that work with subproblems assigned to scenarios. Carøe and Tind [4] showed that applying a primal decomposition method like Benders or L-shaped decomposition to Eq. (2) in general leads to master problems that are governed by non-convex, non-differentiable functions of the same type as the value function of an integer program. Instead, we will in this paper work with a dual decomposition method combined with branch and bound to achieve convergence. This differs from [12, 18] where, although not formally justified, progressive hedging was used for that purpose and proper convergence was empirically observed.

2. Scenario decomposition and Lagrangian relaxation

We consider the following two-stage stochastic program with integer recourse:

$$z = \max\{cx + Q(x): Ax \leqslant b,\ x \in X\} \tag{1}$$

where $Q(x) = \mathbb{E}_\xi \phi(h(\xi) - T(\xi)x)$ and ϕ is given as $\phi(s) = \max\{q(\xi)y: Wy \leqslant s,\ y \in Y\}, s \in \mathbb{R}^{m_2}$. Here c is a known n_1-dimensional vector, A and W are known (rational) matrices of size $m_1 \times n_1$ and $m_2 \times n_2$, respectively, and b is a known m_1-vector. The vector ξ is a random variable defined on some probability space $(\Xi, \mathfrak{F}, P)$ and for each $\xi \in \Xi$, the vectors $q(\xi)$ and $h(\xi)$ and the (rational) matrix $T(\xi)$ have conformable dimensions. The sets $X \subseteq \mathbb{R}^{n_1}_+$ and $Y \subseteq \mathbb{R}^{n_2}_+$ denote restrictions that require some or all of the variables to be integer or binary. Finally, $\mathbb{E}_\xi$ denotes expectation with respect to the distribution of ξ. A *scenario* is a realization of the random variable $(q(\xi), h(\xi), T(\xi))$ corresponding to an elementary event $\xi \in \Xi$. Typically, the distribution of ξ is multivariate. To avoid complications when computing the integral behind $\mathbb{E}_\xi$ we assume that we only have a finite number r of scenarios.

In the following, we use the notation (q^j, h^j, T^j) for the jth scenario having probability p^j, $j = 1, \ldots, r$. When ξ follows a finite distribution problem (1) is equivalent to a large, dual block-angular mixed-integer programming problem. Defining for $j = 1, \ldots, r$ the sets

$$S^j := \{(x, y^j): Ax \leqslant b,\ x \in X,\\ T^j x + W y^j \leqslant h^j,\ y^j \in Y\},$$

the deterministic equivalent can be written

$$z = \max\left\{ cx + \sum_{j=1}^{r} p^j q^j y^j : (x, y^j) \in S^j,\\ j = 1, \ldots, r \right\}. \tag{2}$$

We assume that Eq. (2) is solvable, i.e., feasible and bounded. Here feasibility means that there exists a first-stage solution $x \in X$, $Ax \leqslant b$, such that the feasible regions $\{y^j \in Y: Wy^j \leqslant h^j - T^j x\}$, $j = 1, \ldots, r$, of all the corresponding second-stage problems are non-empty or, in other words, $Q(x) > -\infty$. Boundedness is achieved by requiring feasibility for the dual to the LP-relaxation of Eq. (2).

The fact that Eq. (2) is an integer program of block structure leads to decomposition methods to split it into more manageable pieces. In our approach these pieces will correspond to scenario subproblems. The idea in *scenario decomposition* is to introduce copies $x^1, \ldots, x^r$ of the first-stage variable x and then rewrite Eq. (2) in the form

$$\max\left\{ \sum_{j=1}^{r} p^j (cx^j + q^j y^j) : (x^j, y^j) \in S^j,\\ j = 1, \ldots, r, x^1 = \cdots = x^r \right\}. \tag{3}$$

Here the *non-anticipativity* conditions $x^1 = \cdots = x^r$ state that the first-stage decision should not depend on the scenario which will prevail in the second stage. Of course, there are several equivalent possibilities for expressing this property. Here, we assume that non-anticipativity is represented by the equality $\sum_{j=1}^{r} H^j x^j = 0$ where $H = (H^1, \ldots, H^r)$ is a suitable $l \times n_1 r$ matrix.

The Lagrangian relaxation with respect to the non-anticipativity condition is the problem of finding x^j, y^j, $j = 1, \ldots, r$ such that

$$D(\lambda) = \max\left\{ \sum_{j=1}^{r} L_j(x^j, y^j, \lambda) : (x^j, y^j) \in S^j \right\}, \tag{4}$$

where λ has dimension l and $L_j(x^j, y^j, \lambda) = p^j(cx^j + q^j y^j) + \lambda(H^j x^j)$ for $j = 1, \ldots, r$. The Lagrangian dual of Eq. (3) then becomes the problem

$$z_{\mathrm{LD}} = \min_{\lambda} D(\lambda). \tag{5}$$

The following weak duality result is well known and can be found in, e.g., Nemhauser and Wolsey [13].

Proposition 1. *The optimal value of the Lagrangian dual* (5) *is an upper bound on the optimal value of Eq.* (2). *If for some choice λ of Lagrangian multipliers the corresponding solution (x^j, y^j), $j = 1,\dots,r$ of the Lagrangian relaxation* (4) *is feasible, then (x^j, y^j), $j = 1,\dots,r$ is an optimal solution of problem* (3) *and λ is an optimal solution of Eq.* (5).

The Lagrangian dual (5) is a convex non-smooth program which can be solved by subgradient methods. A major advantage is that it splits into separate subproblems for each scenario,

$$D(\lambda) = \sum_{j=1}^{r} D_j(\lambda),$$

where

$$D_j(\lambda) = \max\{L_j(x^j, y^j, \lambda): (x^j, y^j) \in S^j\}. \tag{6}$$

Each of these r subproblems is a mixed-integer programming problem of size $(m_1 + m_2) \times (n_1 + n_2)$. In contrast, the deterministic equivalent (2) is a mixed-integer program of size $(m_1 + rm_2) \times (n_1 + rn_2)$.

Subgradient methods typically require one function value and one subgradient per iteration as well as a guess of initial multipliers. By convexity, the subgradient $\partial D(\lambda)$ of D at λ is $\partial D(\lambda) = \sum_{j=1}^{r} \partial D_j(\lambda)$ with $\partial D_j(\lambda)$ the set

$$\partial D_j(\lambda) = \mathrm{conv}\{\nabla_\lambda L_j(x, y, \lambda): (x, y) \text{ solves Eq. (6)}\}.$$

Thus $\sum_{j=1}^{r} H^j x^j$ is a subgradient for D where $(x^1, y^1), \dots, (x^r, y^r)$ are optimal solutions of the scenario subproblems (6).

It is well known that due to the integer requirements in Eq. (2), solving Eq. (5) will give an upper bound on z which in general is larger than z. The next proposition provides some insight into why this duality gap arises.

Proposition 2. *The optimal value z_{LD} of the Lagrangian dual* (5) *equals the optimal value of the linear program*

$$\max\left\{\sum_{j=1}^{r} p^j(cx^j + q^j y^j): (x^j, y^j) \in \mathrm{conv}\, S^j,\right.$$
$$\left. j = 1,\dots,r,\ x^1 = \cdots = x^r \right\}. \tag{7}$$

Proof. Theorem 6.2 in [13], p. 327, yields

$$z_{\mathrm{LD}} = \max\left\{\sum_{j=1}^{r} p^j(cx^j + q^j y^j):\right.$$
$$\left. (x, y) \in \mathrm{conv}\ \times_{j=1}^{r} S^j, x^1 = \cdots = x^r \right\}.$$

The assertion then follows from the fact that conv $\times_{j=1}^{r} S^j = \times_{j=1}^{r} \mathrm{conv}\, S^j$. □

The duality gap occurs because the convex hull of feasible solutions to Eq. (3), which is

$$\mathrm{conv}\{(x^1, \dots, x^r, y^1, \dots, y^r): (x^j, y^j) \in S^j,$$
$$j = 1, \dots, r,\ x^1 = \cdots = x^r\},$$

in general, is strictly contained in the set of feasible solutions of Eq. (7). From Proposition 2 it is also clear that the upper bound on z provided by Eq. (5) is not bigger than that obtained by solving the LP-relaxation of Eq. (2) which can be written as

$$\max\left\{\sum_{j=1}^{r} p^j(cx^j + q^j y^j): (x^j, y^j) \in S^j_{\mathrm{LP}},\right.$$
$$\left. j = 1,\dots,r,\ x^1 = \cdots = x^r \right\}$$

where S^j_{LP} arises from S^j by dropping the integer requirements. In fact, our preliminary numerical tests about which we report in Section 4 indicate remarkable improvements of the LP-bound by solving Eq. (5).

The gap can only be closed further by grouping together scenarios in larger blocks, thereby enlarging the size of subproblems (6). For instance, grouping together the ith and jth scenario would yield

$$(\tilde{q}, \tilde{h}, \tilde{T}) = \left((p^i q^i \ \ p^j q^j), \begin{pmatrix} h^i \\ h^j \end{pmatrix}, \begin{pmatrix} T^i & 0 \\ 0 & T^j \end{pmatrix} \right),$$

having probability $p^i + p^j$ and a corresponding scenario subproblem of size $(m_1 + 2m_2) \times (n_1 + 2n_2)$.

Birge and Dempster [2] presents a class of (multi-stage) stochastic programs with integer requirements where the duality gap caused by Lagrangian relaxation of the non-anticipativity conditions vanishes as the number of scenarios tends to infinity. The following formal example shows that for the problems considered in the present paper this cannot be expected in general.

Example 3. Consider the following two-stage stochastic program with integer recourse where only the right-hand side h is random

$$\max\left\{ -3x + \sum_{j=1}^{r} p^j 2y^j : x - \frac{1}{2}y^j \geqslant h^j, 0 \leqslant x \leqslant 1, \right.$$
$$\left. y^j \in \{0,1\}, j = 1,\ldots,r \right\}. \tag{8}$$

We assume that the number r of scenarios is even and that $p^j = 1/r$ for all $j = 1,\ldots,r$. Let ε^j with $0 < \varepsilon^j < \frac{1}{32}$, $j = 1,\ldots,r$, be pairwise distinct real numbers. The right-hand sides h^j are given by $h^j = \varepsilon^j$ if j is even and $h^j = (\frac{1}{4}) - \varepsilon^j$, otherwise.

We will show that, independent of the number r of scenarios, the mentioned duality gap is bigger than $\frac{1}{16}$. To this end we first consider problem (7) for our example. One confirms that $(\frac{1}{2} + \max_j \varepsilon^j, \frac{1}{2}, 1, \ldots, \frac{1}{2}, 1)$ is feasible for that problem such that the maximum is bounded below by $-3\max_j \varepsilon^j > -\frac{3}{32}$. To bound maximum (8) from above let us first consider the case where $y^j = 1$ for some j that is odd. Then $x \geqslant (\frac{3}{4}) - \max_j \varepsilon^j$, and the maximum in Eq. (8) is bounded above by $-(\frac{9}{4}) + 3\max_j \varepsilon^j + 2$ which is less than $-\frac{5}{32}$.

In case $y^j = 0$ for all j that are odd we first consider the situation where $y^j = 0$ for all the remaining j. Then $x \geqslant (\frac{1}{4}) - \max_j \varepsilon^j$, and the maximum in Eq. (8) is bounded above by $-(\frac{3}{4}) + 3\max_j \varepsilon^j$ which is less than $-\frac{21}{32}$.

Finally, if $y^j = 0$ for all j that are odd, and $y^j = 1$ for some remaining j, then $x \geqslant (\frac{1}{2}) + \min_j \varepsilon^j$, and the maximum is bounded above by $-(\frac{3}{2}) - 3\min_j \varepsilon^j + 1$ which is less than $-\frac{1}{2}$.

Altogether, the maximum in Eq. (8) is thus less than $-\frac{5}{32}$ and, hence, the duality gap is at least $\frac{1}{16}$.

When some of the first-stage decisions in the stochastic program (1) are required to be boolean, then there exists a compound representation of the non-anticipativity constraints. For notational convenience we assume that all first-stage decisions are required to be boolean, i.e., that $X = \{0,1\}^{n_1}$. Then we can express non-anticipativity by the single constraint

$$\left(\sum_{j=2}^{r} a_j \right) x^1 = a_2 x^2 + \cdots + a_r x^r, \tag{9}$$

where $a_2,\ldots,a_r$ are positive weights.

A special case of Eq. (9), see also [2], is obtained by letting $a_j = p^j$ for $j = 1,\ldots,r$ for which Eq. (9) can be written as

$$(1 - p^1)x^1 = p^2 x^2 + \cdots + p^r x^r.$$

The Lagrangian with respect to Eq. (9) reads

$$\sum_{j=1}^{r} \tilde{L}_j(x^j, y^j, \mu)$$
$$= \sum_{j=1}^{r} p^j (cx^j + q^j y^j) + \mu \sum_{j=2}^{r} a_j (x^1 - x^j),$$

and the Lagrangian dual

$$z_{\text{LB}} = \min_{\mu} \sum_{j=1}^{r} \max\{\tilde{L}_j(x^j, y^j, \mu) : (x^j, y^j) \in S^j\}. \tag{10}$$

Proposition 4. *The optimal value* z_{LB} *of the Lagrangian dual Eq.* (10) *is greater than or equal to* z_{LD}, *and* z_{LB} *equals the optimal value of the problem*

$$\max\left\{ \sum_{j=1}^{r} p^j (cx^j + q^j y^j) : \right.$$
$$\left. \begin{array}{l} (x^j, y^j) \in \operatorname{conv} S^j, \ j = 1,\ldots,r, \\ \left(\sum_{j=2}^{r} a_j \right) x^1 = a_2 x^2 + \cdots + a_r x^r \end{array} \right\}. \tag{11}$$

Proof. In the same way as in the proof of Proposition 2 it is established that z_{LB} equals the optimal value of Eq. (11). Obviously, the feasible region of Eq. (11) contains that of Eq. (7) which yields $z_{\text{LB}} \geqslant z_{\text{LD}}$. □

The advantage of Eq. (10) is that the number of Lagrangian multipliers is reduced from l to n_1. The number of multipliers affect subgradient procedures in two ways. A small number of parameters give less controllability and the duality gap is increased, viz., Proposition 4. On the other hand, more control parameters mean that a larger space of parameter settings has to be searched and more iterations may be needed.

Our approach can be related to existing techniques in both combinatorial optimization and in stochastic programming. In combinatorial optimization, the idea of creating copies of variables and then relaxing the equality constraints for these variables was introduced as *variable splitting* by Jörnsten et al. [7]. The variable splitting approach was originally applied to optimization problems with a "hard" and an "easy" set of constraints as an alternative to the well-known Lagrangian relaxation approach. The variable splitting method is equivalent to what is called *Lagrangian decomposition* of Eq. (2), by Guignard and Kim [6].

In stochastic programming, non-anticipativity conditions are "hard" since they couple constraints for the different scenarios. For linear problems without integer requirements there is a well-developed theory and methodology for relaxing non-anticipativity constraints. Based on duality results involving augmented Lagrangians, algorithms like the progressive hedging method of Rockafellar and Wets [14] and the Jacobi method of Rosa and Ruszczyński [15] were developed and applied to a variety of problems. As elaborated above duality gaps occur in the presence of integer requirements such that the above methods are no longer formally justified.

Expanding the two-stage decision process behind the stochastic program in Eq. (1) to finite discrete time horizons with arbitrary length leads to multi-stage stochastic programs. We close this section by showing that, from a formal viewpoint, the above approach can be readily extended to multi-stage stochastic programs with integer requirements. From the implementation point of view, however, some work still remains to be done, since problem sizes increase dramatically. As a prototype example we consider the problem

$$\max\{c_1(\xi_1)x_1 + Q_1(x_1) \colon W_1x_1 \leqslant h_1(\xi_1),\ x_1 \in X_1\}, \tag{12}$$

where

$$\begin{aligned} Q_t(x_t) = \mathbb{E}_{\xi_{t+1}|\xi_t} \max\{c_{t+1}(\xi_{t+1})x_{t+1} + Q_{t+1}(x_{t+1}) \colon& \\ T_{t+1}(\xi_{t+1})x_t + Wx_{t+1}& \\ \leqslant h_{t+1}(\xi_{t+1}), x_{t+1} \in X_{t+1}\}& \end{aligned} \tag{13}$$

for $t=1,\dots,T-1$ with $Q_T \equiv 0$. Here $\xi=(\xi_1,\dots\xi_t)$ is a random variable on some probability space $(\Xi, \mathfrak{F}, P)$ and $\mathbb{E}_{\xi_{t+1}|\xi_t}$ denotes expectation with respect to the distribution of ξ_{t+1}, conditional on ξ_t. We assume that ξ_1 is known at time $t = 1$. For all realizations of ξ and time stages, $T_t(\xi_t)$ and W_t are $m \times n$ matrices and $h_t(\xi_t)$ and $c_t(\xi_t)$ are vectors in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively. For convenience we assume that the dimensions are the same for all t. The dynamics of the system as represented in definition (13) of the expected recourse functions is the simplest possible and can be generalized considerably. Finally, the sets $X_t \subseteq \mathbb{R}^n_+$, $t = 1,\dots,T$ include integrality restrictions on the decision variables. We assume that the random vector ξ has finite support $\Xi = \{\xi^1,\dots,\xi^r\}$ and corresponding probabilities $p^1,\dots,p^r$.

In the multi-stage setting scenarios are realizations corresponding to the elementary events in Ξ of the random variables

$$s(\xi) = (s_1(\xi),\dots,s_T(\xi)),$$

where

$$s_t(\xi) = (c_t(\xi_t), T_t(\xi_t), h_t(\xi_t)),\ t = 1,\dots,T.$$

With each scenario vector $s(\xi^j)$ we associate a vector of indeterminates $x(\xi^j)=(x_1(\xi^j),\dots,x_T(\xi^j))$, $j=1,\dots,r$.

Problem (12) can now be restated as a large-scale structured mixed-integer program:

$$\begin{aligned} \max\quad & \sum_{j=1}^{r} p^j[c_1(\xi^j_1)x_1(\xi^j) + \cdots + c_T(\xi^j_T)x_T(\xi^j)] \\ \text{s.t.}\quad & W_1x_1(\xi^j) \leqslant h_1(\xi^j_1), \\ & T_{t+1}(\xi^j_{t+1})x_t(\xi^j) + W_{t+1}x_{t+1}(\xi^j) \leqslant h_t(\xi^j_{t+1}), \\ & x_t(\xi^j) \in X_t \quad \text{for } j = 1,\dots,r \text{ and } t = 1,\dots,T \end{aligned} \tag{14}$$

and the non-anticipativity constraints

$$\begin{aligned} & x_t(\xi^{j_1}) = x_t(\xi^{j_2}) \quad \text{if } s_\tau(\xi^{j_1}) = s_\tau(\xi^{j_2}) \\ & \quad \text{for all } \tau = 1,\dots,t. \end{aligned} \tag{15}$$

These conditions state that two scenarios with the same history until the tth stage should result in the same decisions until this stage. In other words, decisions are only allowed to depend on the past, not on the future. For problems where uncertainties occur at successive time stages the number of scenarios and thereby the number of variables in Eq. (14) grows exponentially with the time horizon. As in the two-stage case we can represent the non-anticipativity conditions (15) by $\sum_{j=1}^{r} H^j x^j = 0$ with a suitable matrix $H=(H^1,\dots,H^r)$, now of dimension $l \times nr(T-1)$, if we assume that (15) altogether comprises l equations. We define the Lagrangian for Eq. (14) as

$$\sum_{j=1}^{r} L_j(x^j,\lambda) = \sum_{j=1}^{r} p^j \sum_{t=1}^{T} c_t(\xi^j_t) x_t(\xi^j) + \sum_{j=1}^{r} \lambda(H^j x^j),$$

where λ has dimension l, and $L_j(x^j,\lambda) = p^j \sum_{t=1}^{T} c_t(\xi^j_t)x_t(\xi^j) + \lambda(H^j x^j)$. Thus the Lagrangian is again separable with respect to scenarios. The Lagrangian dual is then obtained in the same way as above. It becomes a convex non-smooth minimization problem with a total of l variables.

3. A branch and bound algorithm

Lagrangian duality provides upper bounds on the optimal value of problem (3) and corresponding optimal solutions (x^j, y^j), $j=1,\dots,r$, of the Lagrangian relaxation. In general, these *scenario solutions* will not coincide in their x-component unless the duality gap vanishes. We now elaborate a branch and bound procedure for Eq. (1) that uses Lagrangian relaxation of non-anticipativity constraints as bounding procedure. To come up with candidates for feasible first-stage solutions x various heuristic ideas starting from the scenario solutions x^j, $j=1,\dots,r$, can be tried. In the present paper we use the average $\bar{x}=\sum_{j=1}^{r} p^j x^j$, combined with some rounding heuristic in order to fulfill the integrality restrictions. In the following, $\mathscr{P}$ denotes the list of current problems together with associated upper bounds $z_{\mathrm{LD}} = z_{\mathrm{LD}}(P)$. The outline of the algorithm is as follows:

Step 1: *Initialization*: Set $\underline{z} = -\infty$ and let $\mathscr{P}$ consist of problem (1).

Step 2: *Termination*: If $\mathscr{P} = \emptyset$ then the solution $\hat{x}$ that yielded $\underline{z} = c\hat{x} + Q(\hat{x})$ is optimal.

Step 3: *Node selection*: Select and delete a problem P from $\mathscr{P}$, solve the corresponding Lagrangian dual whose optimal value yields the bound $z_{\mathrm{LD}} = z_{\mathrm{LD}}(P)$. If P is infeasible ($z_{\mathrm{LD}} = -\infty$) go to Step 2.

Step 4: *Bounding*: If $z_{\mathrm{LD}}(P) \leqslant \underline{z}$ go to Step 2 (this step can be carried out as soon as the value of the Lagrangian dual falls below $\underline{z}$).

(i) The scenario solutions x^j, $j=1,\dots,r$, are identical: Let $\underline{z}:=\max\{\underline{z}, cx^j + Q(x^j)\}$ and delete from $\mathscr{P}$ all problems P' with $z_{\mathrm{LD}}(P') \leqslant \underline{z}$. Go to Step 2.

(ii) The scenario solutions x^j, $j=1,\dots,r$ differ: Compute the average $\bar{x}$ and round it by some heuristic to obtain $\bar{x}^{\mathrm{R}}$. If $\bar{x}^{\mathrm{R}}$ is feasible then let $\underline{z}:=\max\{\underline{z}, c\bar{x}^{\mathrm{R}} + Q(\bar{x}^{\mathrm{R}})\}$ and delete from $\mathscr{P}$ all problems P' with $z_{\mathrm{LD}}(P') \leqslant \underline{z}$. Go to Step 5.

Step 5: *Branching*: Select a component x_i of x and add two new problems to $\mathscr{P}$ obtained from P by adding the constraints $x_i \leqslant \lfloor \bar{x}_i \rfloor$ and $x_i \geqslant \lfloor \bar{x}_i \rfloor + 1$, respectively (if x_i is an integer component) or $x_i \leqslant \bar{x}_i - \varepsilon$ and $x_i \geqslant \bar{x}_i + \varepsilon$, respectively, where $\varepsilon > 0$ is a tolerance parameter to have disjoint subdomains.

In LP-based branch and bound algorithms for integer programming upper bounds are obtained by relaxing the integrality requirements and feasibility is obtained when a relaxation has an integer optimum. Here we relax the non-anticipativity requirement and feasibility is obtained when the scenario solutions are identical. With mixed-integer variables, the latter is rarely achieved in early steps of the algorithm. Therefore, Step 4(ii) is added where we try to find a feasible solution using the above-mentioned rounding. In the best case this might lead to deletion of subproblems from $\mathscr{P}$ and speed up the branch and bound procedure. It is convenient to introduce a measure for the dispersion of the components in the scenario solutions, which takes into account different ranges of variables, e.g., boolean and continuous variables. Moreover, if the dispersion of component x_i is zero, this should imply non-anticipativity of this component, i.e., $x_i^1 = \cdots = x_i^r$. Standard rules for selecting branching variables and nodes can be adapted to our setting using the dispersion measure, for instance by branching on the component x_i with largest dispersion, selecting the node with the highest norm of dispersion, etc.

In the case of mixed-integer variables some stopping criterion is needed to avoid endless branching on the continuous components. For instance, if we assume that X is bounded and we branch parallel to the coordinate axes, then one may stop after the maximal l_∞-diameter of the feasible sets of the subproblems has fallen below a certain threshold.

Proposition 5. *Suppose that $\{x \in X : Ax \leqslant b\}$ is bounded and that some stopping criterion for the continuous components is employed. Then the branch and bound algorithm above terminates in finitely many steps.*

Notice that a component with zero dispersion will never be branched if the largest-dispersion rule is used for variable selection and the parameter ε in Step 5 above becomes superfluous. The implementation of the node selection and branching rules as well as the stopping criterion will depend on the application at hand. Some preliminary experience is presented in Section 4.

To solve the Lagrangian dual in Step 3 of the algorithm both a non-smooth convex minimization method for solving Eq. (5) and a mixed-integer linear programming solver for the subproblems (6) are needed. In both cases here any such method will do, in principle. However, the number l of non-anticipativity constraints and hence the dimension of λ in the Lagrangian dual (5) increases with the dimension of the first-stage vector x (for two-stage problems) and the number of scenarios used to represent uncertainty. Furthermore, the mixed-integer subproblems (6) have to be solved a number of times depending on the number of iteration steps to solve Eq. (5) and, again, the number of scenarios. Therefore, it is advisable to use advanced methodology for solving Eqs. (5) and (6).

In our implementation we used NOA 3.0 [8], a code by Kiwiel based on his proximal bundle method [9], for solving the Lagrangian dual. In contrast to simple subgradient updates this algorithm employs both proximal point and cutting plane methodology to achieve global convergence in theory and encouraging numerical results in practice. For further details including relations with other non-smooth convex minimization methods we refer to [9]. The mixed-integer subproblems (6) provide function values and subgradients which the proximal bundle method needs at each iteration. For their solution we use the CPLEX 4.0 Callable Library [5], i.e., state-of-the-art software for LP-based branch and bound.

The above branch and bound algorithm readily extends to the multi-stage case. Again Lagrangian relaxation of non-anticipativity constraints can be embedded as bounding procedure. Since non-anticipativity constraints involve variables from all but the final time stage, branching in the multi-stage case has to concern the variables $x_1, \ldots, x_{T-1}$ instead of only x_1 in the two-stage case. Together with the increased dimension of the Lagrangian dual this more expensive branching is a main source of increased computational effort such that from the implementation point of view many open problems remain in the multi-stage case.

4. Numerical examples

We have implemented the branch and bound algorithm of Section 3 for the two-stage case using the constraints $x^1 = x^2$, $x^1 = x^3, \ldots, x^1 = x^r$ to represent non-anticipativity.

At each node of the branching tree we chose to branch on the component x_i for which the dispersion $\max_j x_i^j - \min_j x_i^j$ was largest. For node selection we chose the node with the largest l_∞-norm of dispersions. To obtain good lower bounds we used this rule intertwined with the Best-Bounds rule. The experiments were carried out on a DEC Alpha workstation with 192 MB memory and 266 MHz processor.

Example 1. The following stochastic program was adapted from [16]:

$$\max\{\tfrac{3}{2}x_1 + 4x_2 + Q(x_1,x_2) : 0 \leqslant x_1, x_2 \leqslant 5 \text{ and integer}\} \tag{16}$$

where $Q(x_1,x_2)$ is the expected value of the multi-knapsack problem

$$\begin{array}{ll} \max & 16y_1 + 19y_2 + 23y_3 + 28y_4 \\ \text{s.t.} & 2y_1 + 3y_2 + 4y_3 + 5y_4 \leqslant \xi_1 - x_1, \\ & 6y_1 + y_2 + 3y_3 + 2y_4 \leqslant \xi_2 - x_2, \quad y_i \in \{0,1\}, \\ & \quad i = 1, \ldots, 4 \end{array}$$

and the random variable $\xi = (\xi_1, \xi_2)$ is uniformly distributed on $\Xi = \{(5,5),(5,6),\ldots,(5,15),(6,5),\ldots,(15,15)\}$, giving a total of 121 scenarios. A first

Table 1
Table of results for SIZES-problems

Problem	Best solution	CPU-time (s)	Lower bound
SIZES3	224580.3	522	224358.1
SIZES5	224626.0	232	224351.0
SIZES10	224714.4	635	224347.4

attempt at solving (16) would be to solve directly its deterministic equivalent which is an integer program with 484 binary variables, 2 integer variables and 242 constraints.

Applying the CPLEX 4.0 MIP solver, the problem could not be solved to optimality in 300.000 nodes, yielding an optimality gap of more than 10%. Using our branch and bound algorithm the problem could be solved in 6.9 s CPU-time, yielding the optimal solution $x=(0,4)$ and corresponding value $z=62.29$. Notice that the scenario subproblems are very small, so better runtimes may be achieved by grouping together scenarios.

Example 2. To compare the behavior of our algorithm with problems from the literature having larger second stages, we consider a family of two-stage mixed-integer minimization problems analyzed in [12]. The problems SIZES3, SIZES5 and SIZES10 have 3, 5 and 10 scenarios, respectively, and the scenario subproblems in each stage have 10 boolean variables, 65 bounded continuous variables and 31 constraints. Randomness occurs only in the right-hand side of the second-stage problem.

The computational results are summarized in Table 1. The second column shows the time after which the best-feasible solutions were found and the third column shows the lower bounds obtained after 1000 s of CPU-time, where the test runs were stopped. Contrary to the method in [12], we can estimate the feasible solutions found to be within 0.2% of the optimum. The Lagrangian dual provides considerably better lower bounds than the LP-relaxation. For our test runs we used 10^{-3} as optimality tolerance in NOA which gave at the root nodes a duality gap of only 0.2–0.3%. The LP-relaxation, however, gives a duality gap of 2.0–2.1%. Notice that a smaller optimality tolerance in NOA will produce better bounds but is also more time consuming.

It should be noted that the size of the duality gap indicates that the scenario solutions have almost identical first-stage components. Hence, we calculated the *value of the stochastic solution* (VSS), see Birge [1], which measures the value of using a stochastic model instead of a deterministic model. For all three problems, the VSS was less than 0.8%, which means that the randomness has little influence on the optimal first-stage solution.

Acknowledgements

We are grateful to Krzysztof C. Kiwiel for permission to use the NOA 3.0 package and to A. Løkketangen and D.L. Woodruff for providing us with data for the SIZES-problems. Moreover we wish to thank an anonymous referee for comments that helped to improve the presentation of the paper.

References

[1] J.R. Birge, The value of the stochastic solution in stochastic linear programs with fixed recourse, Math. Programm. 24 (1982) 314–325.

[2] J.R. Birge, M.A.H. Dempster, Stochastic programming approaches to stochastic scheduling, J. Global Optim. 9 (1996) 417–451.

[3] J.R. Birge, F.V. Louveaux, Introduction to Stochastic Programming, Springer, New York, 1997.

[4] C.C. Carøe, J. Tind, L-shaped decomposition of two-stage stochastic programs with integer recourse, Math. Programm. 83 (1998) 451–464.

[5] CPLEX Optimization, Inc., Using the CPLEX Callable Library, CPLEX Optimization, 1989–1995.

[6] M. Guignard, S. Kim, Lagrangean decomposition: a model yielding stronger bounds, Math. Programm. 39 (1987) 215–228.

[7] K.O. Jörnsten, M. Näsberg, P.A. Smeds, Variable splitting-a new lagrangean relaxation approach to some mathematical programming models, Department of Mathematics Report LiTH-MAT-R-85-04, Linköping Institute of Technology, Sweden, 1985.

[8] K.C. Kiwiel, User's Guide for NOA 3.0: A Fortran Package for Convex Nondifferentiable Optimization, Systems Research Institute, Polish Academy of Sciences, Newelska 6, 01-447 Warsaw, Poland.

[9] K.C. Kiwiel, Proximity control in bundle methods for convex nondifferentiable optimization, Math. Programm. 46 (1990) 105–122.

[10] W.K. Klein Haneveld, L. Stougie, M.H. van der Vlerk, An algorithm for the construction of convex hulls in simple integer recourse programming, Ann. Oper. Res. 64 (1996) 67–81.

[11] G. Laporte, F.V. Louveaux, The integer L-shaped method for stochastic integer programs with complete recourse, Oper. Res. Lett. 13 (1993) 133–142.

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

[13] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, Wiley-Interscience, New York, 1988.

[14] R.T. Rockafellar, R.J.-B. Wets, Scenarios and policy aggregation in optimization under uncertainty, Math. Oper. Res. 16 (1991) 1–23.

[15] C.H. Rosa, A. Ruszczyński, On augmented Lagrangian decomposition methods for multistage stochastic programs, Ann. Oper. Res. 64 (1996) 289–309.

[16] R. Schultz, L. Stougie, M.H. van der Vlerk, Solving stochastic programs with integer recourse by enumeration: a framework using Gröbner basis reductions, Math. Programm. 83 (1998) 229–252.

[17] L. Stougie, M.H. van der Vlerk, Stochastic integer programming, Ch. 9, in: M. Dell'Amico, F. Maffioli, S. Martello (Eds.), Annotated Bibliographies in Combinatorial Optimization, Wiley, New York, 1997, pp. 127–141.

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