STOCHASTIC INTEGER PROGRAMMING SOLUTION THROUGH A CONVEXIFICATION METHOD

Santiago Cerisola, Jesus M. Latorre, Andres Ramos

Escuela Técnica Superior de Ingeniería ICAI, Universidad Pontificia Comillas,
Alberto Aguilera 23, 28015 Madrid, Spain, santiago.cerisola@iit.upcomillas.es

In this paper we present a solution method for stochastic integer problems. The method is a Benders-type algorithm that sequentially approximates the nonconvex recourse functions defined by the second stage subproblems. The presented convexification takes into account the domain that is induced by the collection of tender variables. The method is applied to a broad collection of stochastic integer programming problems taken from the literature and a summary of the numerical results is presented.

Keywords: Benders decomposition, lagrangean relaxation

JEL Classification codes: C61, C63

1 INTRODUCTION

Stochastic programming (SP) deals with mathematical programming problems where some of the input data are random parameters (Birge and Louveaux 1997). The method employed to solve an SP problem highly depends on the underlying characteristics of the random parameters. SP problems involving random parameters whose probability distributions are approximated with continuous functions are mainly solved via sampling and simulation techniques. On the other hand, SP problems where probability distributions are approximated by discrete functions may in principle be solved with conventional optimization software via the formulation of their deterministic equivalent problem.

Many SP problems may be formulated as multiperiod or multistage problems, where each period indicates a moment in which a decision has to be made. The natural way of representing uncertainty in this type of problems when probability distributions are approximated with discrete functions is by a scenario tree (Dupacová, Consigli et al. 2000).

The introduction of stochasticity into a mathematical programming problem and its solution through a deterministic equivalent problem greatly increases the computational effort required. Not surprisingly, decomposition techniques appear as alternative or complement strategies to the direct solution of these problems (Ruszczynski 1997). For linear cases with two decision stages, Benders' decomposition technique is the most widely used (Benders 1962; Van Slyke and Wets 1969). The introduction of three or more decision stages leads to the immediate generalization of the method to nested decomposition schemes (Morton 1993). When stochasticity is introduced in the form of a scenario tree, the decomposition method can also be extended and used with a monocut version or with a multicut version (Birge and Louveaux 1988).

When stochasticity is introduced through continuous probability distribution functions, the deterministic version is extended to accept simulation and sampling in the decomposition algorithm (Birge and Louveaux 1997).

The introduction of integer variables in such a decomposition scheme complicates the development of a solution method. The original two-stage or L-shaped method (Benders 1962) formulates the first-stage problem so that it comprises the collection of integer variables whereas the second-stage problem deals with the rest of decision variables with the first-stage integer decisions fixed. The method exploits the linearity of the second-stage problem in order to outer approximate the convex recourse function, which represents the dependence of the second-stage objective function with respect to the first-stage decisions. However, when integer variables are included in the second-stage problem this recourse function is in general non convex and non continuous. Many efforts have been done to overcome this difficulty. The integer L-shaped method of (Laporte and Louveaux 1993), uses an adequate expression to outer approximate the recourse function for problems with only 0-1 first-stage decision variables and mixed-integer second-stage decision variables. Its disadvantage is that not all problems fit within such structure. The specific method developed by (Van der Vlerk 1995) for simple integer recourse problems exploits the pseudo-convexity properties of the recourse function for this type of problems including the extension to nested cases when each subproblem presents a simple integer recourse structure. Again, not every problem can be formulated in a natural way as a simple integer recourse problem. Other extensions of the method to deal with nonconvex recourse functions that have been developed are oriented to solve problems with general structures. For example, the branch-and-bound subproblem solution of (Birge and Louveaux 1997) computes one Benders' cut at each terminal node of the ramification tree. This procedure has the purpose of treating the collection of cuts generated in each iteration by means of a disjunctive logic constraint: only one of these cuts is allowed to be active in subsequent iterations. The combination of branch and bound techniques with decomposition methods has provided successful results in (Ahmed, Tawarmalani et al. 2004). This later B&B is limited to problems with pure integer first stage variables. The method transforms the space of tender variables with the purpose of having the discontinuities of the recourse function orthogonal to the variable axes.

Other authors such as (Olaf E. Flippo 1993) use generalized duality instead of linear duality with the purpose of approximating the recourse function. For example (Carøe & Tind, 1998) generate a non-continuous approximation of the recourse function by using subbadditive theory in the solution of the second-stage problem. However, the technique does not lead to an efficient algorithm to solve this type of problems. They are theoretical extensions to the linear case. For example, the cuts generated by (Carøe & Tind, 1998) are neither convex nor continuous, which complicates their use with algorithmic purposes.

Convexification is a natural approach to address the solution of these problems. Generalized Benders' decomposition (GBD), originally proposed by (Geoffrion 1972), uses nonlinear duality to approximate the convexification of the recourse function. Our paper proposes an extension of GBD that considers the domain of the recourse function induced by the collection of tender variables. Convexification procedures have also the goal of obtaining the convex hull of the second stage feasible region. In (Sen and Kothari 1998), disjunctive programming is used into the second stage of the L-shape method to approximate this convex hull. The method presented in (Sen and Kothari 1998) has the advantage that the disjunctive cuts obtained may be shared among the second-stage individual scenario subproblems.

Alternative to L-shaped methods, which profit from the block structure of decision variables, Lagrangean methods exploit the block structure of constraints to eliminate those that complicate the solution of the problem (Geoffrion 1974). Stochastic programming is a suitable field to apply these Lagrangean methods. The usual approach is to derive an extended formulation of the stochastic problem that includes copies of the decision variables for each scenario and explicitly incorporates the formulation of the non-anticipativity constraints. These constraints force the decisions taken at each period to be independent of future realizations of the uncertainty. Non-anticipativity constraints introduce a link between the copies of the decision variables corresponding to different scenarios. The Lagrangean relaxation of these constraints leads to a Lagrangean subproblem that is separable into individual subproblems, each one corresponding to one scenario. This technique is commonly known as scenario decomposition or progressive hedging algorithm when a regularized term is added to the subproblem objective function (Rockafellar and Wets 1991). A disadvantage of these methods appears in the dimension of the lagrangean function, which is usually extremely large for the application of any outer approximation method. Another disadvantage is that a primal solution is usually not available after the Lagrangean algorithm, and B&B or heuristic methods are frequently required to postprocess the given solution.

In this paper we present a Benders-type algorithm that approximates the lower convex envelope of the nonconvex recourse function. This convexification considers the domain of the recourse function imposed by the tender variables. Additionally, the paper also presents the application of the method to a battery of test-sets problem obtained from the literature (stoprog.org). In the application of the method, a sequential approach has been adopted. The code developed for obtaining numerical results generates computationally cheaper approximations of the recourse function prior to other more computationally expensive ones. This sequential method stems from our experience with the solution of mixed-integer programming (MIP) problems. The closer to the optimal solution the algorithm is, the more effort is required to improve the achieved solution. The sequential approach makes it possible to stop the algorithm as soon as a certain tolerance has been reached. This idea is particularly interesting in a stochastic integer programming (SIP) environment, where the difficulty of obtaining a solution for an integer programming (IP) problem is combined with the curse of dimensionality typical of stochastic programming (SP) problems.

The rest of the paper is organized as follows. Section 2 briefly reviews Benders' decomposition or L-shaped method and presents the proposed convexification procedure. Section 3 summarizes the numerical results and tests the algorithm efficiency. Finally, section 4 presents the main conclusions of this research.

2 L-SHAPED DECOMPOSITION AND CONVEXIFICATION

The method we are about to present may be formulated for multistage stochastic problems. However, for the shake of simplicity, we describe the algorithm for a two-stage problem, pointing out those parts where a particular nested scheme should be necessary.

2.1 The L-Shaped Method

Benders' or L-shaped decomposition considers two-stage optimization problems that can be formulated in the following form:

$$(P) \qquad \begin{array}{l} \min cx + qy \\ Tx + Wy = h \\ x \in X, y \in Y \end{array} \tag{1}$$

where x represents first-stage decisions and y comprises second-stage variables whose feasible regions are respectively given by $X = \left\{ A_1 x \le a_1, x \in \mathbb{R}_+^{n_1} \right\}$ and $Y = \left\{ A_2 y \le a_2, y \in \mathbb{R}_+^{n_2} \right\}$. The solution of problem (P) is equivalent to the solution of the following master problem (MP).

$$(MP) \qquad \min \left\{ cx + Q(x), x \in X \right\} \tag{2}$$

where $Q(x)$ is the recourse function which is defined by the following subproblem (SP_x):

$$(SP_x) \qquad Q(x) = \min \left\{ qy, Wy = h - Tx, y \in Y \right\} \tag{3}$$

The L-shaped algorithm replaces the recourse function $Q(x)$ in the master problem (MP) by a partial description that is updated as the algorithm proceeds. This description of the recourse function is derived by application of linear duality. Indeed, the recourse function $Q(x)$

may also be represented as $\max_{i\in I}\left\{\pi^i(h-Tx)+\rho^i a_2\right\}$, where $\left\{(\pi^i,\rho^i)_{i\in I}\right\}$ is the collection of extreme dual solutions of problem (3). Observe that this representation of the recourse function is based on linear cuts. This outer approximation of the recourse function is complemented in the decomposition algorithm by the outer approximation of the first-stage feasibility region, which is given by the collection of first-stage solutions such that Benders' subproblem (SP_x) is feasible. This feasibility region can be represented as $\left\{x/0\geq\tilde{\pi}^j(h-Tx)+\tilde{\rho}^j a_2\right\}_{j\in J}$, where $\left\{(\tilde{\pi}^j,\tilde{\rho}^j)_{j\in J}\right\}$ is the collection of extreme dual solutions that result from the minimization of infeasibilities of (SP_x), (Birge and Louveaux 1997).

An alternative formulation for Benders' cuts can be derived that will prove useful later on. Let π^* and $\theta^*_{x_0}$ be the optimal dual value and optimal solution of a feasible subproblem (SP_{x_0}) when a certain first-stage solution x_0 has been proposed. Then, the following is a lower bound for the recourse function:

$$\begin{aligned} Q(x) &\geq \pi^*(h-Tx)+\rho^* a_2=\pi^*(h-Tx+Tx_0-Tx_0)+\rho^* a_2=\\ &=\pi^*(h-Tx_0)+\rho^* a_2+\pi^*(-Tx+Tx_0)=\theta^*_{x_0}+\pi^* T(x_0-x) \end{aligned} \tag{4}$$

The decomposition algorithm solves in each iteration a relaxed master problem (RMP) given by

$$(RMP)\qquad \begin{aligned} &\min cx+\theta\\ &0\geq\theta^j+\tilde{\pi}^j T(x_0^j-x)\quad j\in J'\\ &\theta\geq\theta^i+\pi^i T(x_0^i-x)\quad i\in I'\\ &x\in X \end{aligned} \tag{5}$$

where I' is a subset of I and correspond to the set of optimality cuts, J' is a subset of J and correspond to the set of feasible cuts and x_0^i is the master proposal that generated cut *i*.

Each iteration of the method starts with the solution of (RMP) and the proposal of a first-stage solution, x_0. This first-stage solution is then used to evaluate the recourse function by solving the corresponding subproblem (SP_{x_0}). The description of the recourse function in (RMP) is enhanced with an optimality cut i in case of subproblem feasibility. In the other case, the feasibility region of (RMP) is constrained with a feasibility cut j. Simultaneously, the algorithm computes a lower and an upper bound for the objective function of (P) and stops when the relative difference is less than an appropriate tolerance.

Step 0	Set $i' = j' = 0$. Set $\theta \equiv 0$ at the initial iteration
Step 1	Solve (RMP) and obtain solution x_0 and lower bound $\underline{z} = \upsilon(RMP)$
Step 2	Solve (SP_{x_0})
	If (SP_{x_0}) is infeasible set $J' = J'+1$, and obtain $\tilde{\pi}^{j'}$
	If (SP_{x_0}) is feasible set $I' = I'+1$, and obtain $\pi^{i'}$
	Compute upper bound $\bar{z} = cx_0 + \upsilon(SP_{x_0})$
Step 3	(stopping rule)
	If $\left(\bar{z} - \underline{z}\right)\big/\bar{z} < tol$ stop, x_0 is optimal solution, else go to Step 1

Algorithm 1. L-Shaped method.

The two-stage L-shaped method is immediately extended to multistage problems via nested decomposition and to stochastic problems with the use of the multicut or monocut version of the method.

2.2 An extension for non-convex recourse functions

The introduction of integer requirements within the second-stage decision variables turns non-convex the recourse function (3). A clear exposition of the convexification of the recourse function can be given through the concept of graph and epigraph for the second stage-subproblem. Let

$$G = \left\{(r, r_0) / \exists y \in Y \text{ with } Wy - h = r \text{ and } qy = r_0\right\} \tag{6}$$

be the graph of the second stage subproblem. The second stage subproblem, for a fixed first stage value x_0, may now be reinterpreted as finding a point $(-Tx_0, r_0)$ in G with minimum ordinate. Similarly, consider

$$EpiG = \left\{(r, r_0) / \exists y \in Y \text{ with } Wy - h = r \text{ and } qy \le r_0\right\} \tag{7}$$

be the epigraph of the second stage subproblem. Any linear expression that outer-approximates the epigraph of the second stage may turn into a valid cut to approximate the recourse function. A simple way of deriving a valid inequality for the lower convex envelope of the epigraph (7) is achieved by considering a multiplier value λ and evaluating the dual function $\omega(\lambda)$, defined as:

$$\omega(\lambda) = \min_{(r, r_0)} \left\{\lambda r + r_0 / (r, r_0) \in EpiG\right\} \tag{8}$$

The optimal solution of above problem provides a valid cut that approximates the lower convex envelope of the epigraph given as:

$$r_0 \ge \omega(\lambda) - \lambda r \tag{9}$$

It is important to notice that the expression (7) of the epigraph does not take into account the fact that the values that r can take are restricted to the set $\{-Tx, x \in X\}$. We suggest to improve the above construction of a valid cut by considering $\{r = -Tx, x \in X\}$, and redefining the dual function as

$$\omega(\lambda) = \min_{x,y} \{-\lambda Tx + qy, x \in X, y \in Y\} \tag{10}$$

This problem generates a linear cut that outer approximates the nonconvex recourse function whose expression is

$$Q(x) \geq \omega(\lambda) + \lambda Tx \tag{11}$$

The choice of the multiplier λ that enters in above expression may be conditioned by the subproblem that needs to be solved at any given iteration of the L-shaped method. For a given first stage proposal x_0, a possibility is to find the multiplier that returns the maximum of the lower bounds provided by (11). The dual problem (D) is defined as the optimization problem that selects the optimal multiplier.

$$(D) \qquad \max_{\lambda} \left(\min \{-\lambda Tx + qy, x \in X, y \in Y\} + \lambda Tx_0 \right) \tag{12}$$

The reader may appreciate the similarity of optimality cut (11) with that of expression (4). Linearizing around the value x_0 and after some algebra we find

$$Q(x) \geq \omega(\lambda) + \lambda Tx_0 - \lambda Tx_0 + \lambda Tx = \theta_{x_0} - \lambda T(x_0 - x) \tag{13}$$

Expression (13) recovers the traditional result that relates the dual variable of a linear programming problem with the opposite of the optimal multiplier that the nonlinear duality theory provides. This result is of remarkable interest in the development of an efficient algorithm for realistic stochastic integer programming problems. We will take this issue again in the next section.

We remark that above lines have assumed that the second-stage mixed-integer problem is feasible for any given first-stage value. This assumption is not necessary for the correct behavior of the algorithm. In case of infeasibility of the second stage problem (for a given first stage value), the convexification procedure can be carried out replacing the objective function by that of minimization of infeasibilities. This convexification creates a feasibility cut that eliminates the first stage proposal out of the collection of first–stage solutions.

2.3 Description of the algorithm

The algorithm we propose for stochastic integer programming problems considers stochasticity given by discrete distributions. For multistage programs, the method assumes that random parameters adopt a tree-shape form. The goal is to find the first stage solution such that the total expected cost is minimized:

$$\min \left\{ cx + E_{\omega\in\Omega} Q(x,w), x \in X \right\} \tag{14}$$

with

$$Q(x,w) = \min \left\{ q(\omega)y, W(\omega)y = h(\omega) - Tx, y \in Y \right\} \tag{15}$$

For the particular case we are about to solve, i.e. stochastic parameters discretely distributed, problem (14) can be formulated as

$$\min \left\{ cx + \sum_{\omega\in\Omega} p_\omega Q(x,w), x \in X \right\} \tag{16}$$

The L-shaped method, reviewed in section 2.1 for the deterministic counterpart to (14) in continuous variables, is applied in a natural way to the L-shaped decomposition of problem (16) . A master problem (MP) comprises the collection of variables that represent first-stage decisions and partial approximations to the lower convex envelope of the recourse functions.

$$(MP) \qquad \begin{aligned} & \min cx + \sum_{\omega\in\Omega} p_\omega \theta_\omega \\ & 0 \ge \theta_\omega^j + \tilde{\pi}_\omega^j T(\omega)(x_0^j - x) \quad j \in J'(\omega) \\ & \theta_\omega \ge \theta_\omega^i + \pi_\omega^i T(\omega)(x_0^i - x) \quad i \in I'(\omega) \\ & x \in X \end{aligned} \tag{17}$$

This master problem proposes first-stage decisions that modify the right-hand-side of each scenario subproblem (15). In (17) we have adopted the notation used for optimality and feasibility cuts of the linear problem. This adoption has been done in order to maintain clarity, and because the optimality and feasibility cuts obtained for the linear relaxation of the individual scenario subproblems are valid when imposing integrality requirements to the decision variables.

The L-shaped method we present in this section has been applied in a sequential way. It is defined as cut refinement method. It executes Benders' algorithm computing Benders' cuts in different ways as the algorithm proceeds. Easier cuts are calculated first and more expensive cuts are calculated later. The method is organized in three phases where each phase is characterized by the way of computing the Benders' cuts.

2.3.1 Phase 1

In Phase 1 we remove integrality requirements from the subproblems and we apply the traditional linear Benders' decomposition algorithm. Hence, in forward and backward[1] passes we solve the linear relaxation of the (MP) problem. In each iteration, the accuracy of the linear solution achieved is calculated as the relative difference between a linear upper bound and a linear lower bound, as usual in Benders' algorithm. The linear lower bound is given by the objective value of the relaxed master problem whereas the linear upper bound is obtained by evaluating the objective function of the complete problem with the latest solution. Phase 1 ends when the relative difference between these two bounds is smaller than a certain tolerance.

2.3.2 Phase 2

In phase 2 we reincorporate the integrality requirements that we removed in phase 1. When we traverse forward the problems, we solve their MIP version (this also holds for phase 3). In the backward pass, we generate a linear cut that approximates the recourse function. In order to reduce the computational cost of solving the dual problem (12), we choose the multipliers that will enter in the evaluation of the dual function (10). Our selection corresponds to the opposite of the dual variables of the subproblem linear relaxation.

In a two-stage case, this technique shifts the linear Benders cut until it touches the recourse function. The cut obtained will in general not be tangent to the lower convex envelope at the point proposed by the master problem but, in any case, it is a valid cut and it is stronger than the linear Benders cut. This improvement has the computational cost of solving a MIP subproblem instead of a LP subproblem.

When dealing with integer variables, the stopping rule of the linear Benders algorithm is no longer valid. There is not guarantee that the lower bound provided by the resolution of the root problem will be equal to the evaluation of the objective function at the optimal solution. For this reason, the stopping criterion is modified and this phase ends when the difference of the primal values obtained in two consecutive iterations is lower than a specified tolerance.

---

[1] We are considering in the description of the method the natural extension to nested problems. A forward pass indicates subproblem resolutions that are oriented to obtain primal proposals for descendent subproblems. A backward pass indicates subproblem resolutions oriented to generate approximations of the recourse function lower convex envelope.

2.3.3 Phase 3

In this phase, we solve the MIP version of the subproblems and obtain primal values for their descendents. In backward passes, we solve the dual problem (12) of each subproblem to find the optimality cut that better approximates the convex envelope of the recourse functions at the proposal given by its ancestor master problem. The cuts calculated in this phase are certainly harder to compute than those of phase 2. For this reason, in the code developed with the purpose of testing the method, we control whether it is necessary to solve the dual problem or not. The duality gap is computed when a primal solution is available. This duality gap is given by the difference of the evaluation of the recourse function (computed in the forward pass), and the lower bound provided by the lifting procedure of phase 2. Thus, the optimization of the dual function in only invoked when the optimality gap is greater than a given tolerance.

In this phase, the stopping rule maintains the criterion of phase 2.

2.3.4 Summary and observations

In the cut refinement method, as soon as we are in the phase 2 we have feasible solutions for the complete problem and we can obtain upper bounds for the solution by evaluating the objective function of the complete problem at any of these feasible solutions. Additionally, a lower bound is provided by the master problem.

The method may be stopped at any phase if the tolerance required for the solution is reached. This is useful because it permits avoiding phase 3, which is dramatically more time consuming than the previous ones.

A major drawback of this sequential method that must be highlighted is the possibility of having an LP feasible subproblem that turns out to be MIP infeasible. Although this rarely happens, in such case the algorithm triggers the maximization of the dual problem (12), with the objective function $q(\omega)y$ replaced by that of minimization of infeasibilities, which is computationally expensive.

The following table summarizes the sequential cut refinement method.

	Forward Solution	Backward Solution
Phase 1	LP	LP
Phase 2	MIP	LP + Lagrangean Subproblem
Phase 3	MIP	Max-Min Subproblem

Table 1. L-shaped and convexification method.

2.3.5 Convergence of the method

The method converges to a first stage solution that minimizes first stage cost and the expected cost of the lower convex envelope functions. However, this is not the total expected cost, which is the sum of the first stage cost and the expected cost of the second stage objective function. It is clear that the convexification of the sum of nonconvex functions is not the sum of the individual convexifications, so that it cannot be asserted that the method converges to the optimal solution of the problem (a short example is presented in the next section). However, for some problems, the approximation provided by the method is good enough, which motivates the use of the method for large-scale problems.

3 NUMERICAL RESULTS

We have applied the proposed method to a collection of examples provided in the Stochastic Programming Community homepage (stoprog.org) as well as stochastic models arising from the field of power systems operation planning. These problems present a suitable staircase structure that gives the possibility of using the L-shaped algorithm to obtain the optimal solution. The collection of tested problems is now presented.

Ex1. This example is taken from (Ahmed, Tawarmalani et al. 2004) and is the problem:

$$\begin{aligned} \min \ & -1.5x_1 - 4x_2 + E[Q(x_1, x_2, \omega_1, \omega_2)] \\ & x_1, x_2 \in [0,5] \cap \mathbb{Z} \end{aligned}$$

with

$$\begin{aligned} Q(x_1, x_2, \omega_1, \omega_2) = \min & -16y_1 - 19y_2 - 23y_3 - 28y_4 \\ \text{s.t. } & 2y_1 + 3y_2 + 4y_3 + 5y_4 \le \omega_1 - x_1 \\ & 6y_1 + \ y_2 + 3y_3 + 2y_4 \le \omega_2 - x_2 \\ & y_1, y_2, y_3, y_4 \in \{0,1\} \end{aligned} \tag{18}$$

and (ω_1, ω_2) is considered to be uniformly distributed on $\Omega \subseteq [0,5] \times [0,5]$. In similarity with (Ahmed, Tawarmalani et al. 2004) five instances of 4, 9, 36, 121 and 441 scenarios have been tested. Computational results are presented in table 2.

Ex2. This example is a variant of Ex1 and consists of Ex1 where the integrality requirements have been removed from the first stage decision variables. See table 3.

Ex3. This example is a variant of Ex1 with a different technology matrix T. The recourse function of (18) is replaced by

$$\begin{aligned} Q(x_1, x_2, \omega_1, \omega_2) = \min & -16y_1 - 19y_2 - 23y_3 - 28y_4 \\ \text{s.t. } & 2y_1 + 3y_2 + 4y_3 + 5y_4 \le \omega_1 - \frac{1}{3}x_1 - \frac{2}{3}x_2 \\ & 6y_1 + \ y_2 + 3y_3 + 2y_4 \le \omega_2 - \frac{2}{3}x_1 - \frac{1}{3}x_2 \\ & y_1, y_2, y_3, y_4 \in \{0,1\} \end{aligned}$$

and the five stochastic instances of Ex1 have also been tested. See table 4.

Sslp. This example arises in stochastic server location problems. The problem presents pure binary first-stage variables and mixed-binary second stage variables. The collection of instances provided in the web page are different instances named Sslp_*m*_*n*_*s*, where *m* is the number of potential server locations, *n* is the number of potential clients, and *s* is the number of scenarios. This notation is maintained in table 5, which presents the numerical results obtained.

Sizes. This example is a two-stage multiperiod stochastic integer program that arises in product substitution applications. The problem has mixed integer variables in both stages. Three stochastic instances of the problem are available and the numerical results achieved with our method are given in table 6.

Dcap. This problem comes from a dynamic capacity acquisition and allocation application described in (Ahmed and Garcia 2002). This problem is formulated as a two-stage integer programming problem where the first-stage decisions consist on determining the capacity that needs to be expanded of a given resource (or collection of resources) all over a time scope of T periods. The second stage problem assigns the available resources to the tasks that have to be realized. This assignment is modeled with binary variables, leading to a pure integer second stage subproblem.

The problem is stochastic because the expansion decisions have to be carried out (for the entire horizon) before the disclosure of uncertainty. Uncertainty that modifies the cost of assigning resources to different tasks as well as the amount of resource that is required by the different tasks. This problem presents a suitable staircase structure for the application of L-shaped methods. Twelve stochastic instances of the model have been solved, see table 7.

Muc. This example is from the field of power systems operation planning (Cerisola, Baíllo et al. 2005). The problem is a stochastic unit commitment model for a power generation company that takes part in an electricity spot market. The relevant feature of this model is its detailed representation of the spot market during a whole week, including seven day-ahead market sessions and the corresponding adjustment market sessions. This representation takes into account the influence that the company’s decisions exert on the market clearing price by means of a residual demand curve for each market session. Uncertainty is introduced in the form of several possible spot market cases for each day, which leads to a weekly scenario tree. The model also represents in detail the operation of the company’s generation units, as usual in unit commitment models. The proposed unit commitment model leads to large-scale mixed integer maximization problem, with mixed-integer variables at each of the periods of the problem. The staircase structure of the matrix can be appreciated in figure 1, showing the constraints for the corresponding seven days.

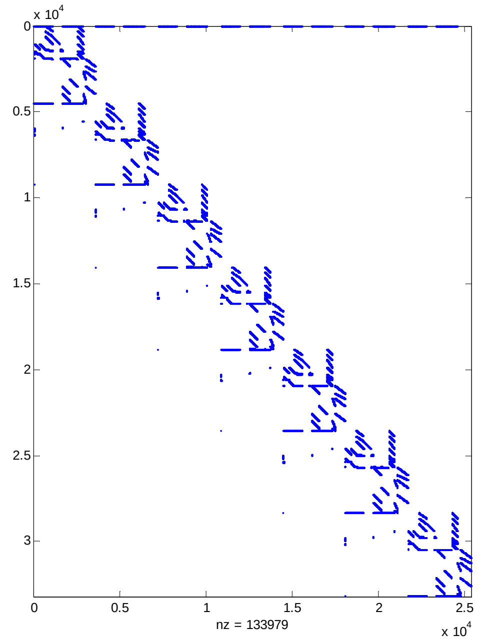

Figure 1. Matrix structure of a single-scenario 7 periods market unit commitment problem

A difference with the examples so far presented is that the introduction of stochasticity for this market unit commitment problem forces the solution method to be applied in their nested version. An eight scenarios problem has been solved. The subtree structure that splits the deterministic equivalent problem for the decomposition method has been chosen to be that of figure 2.

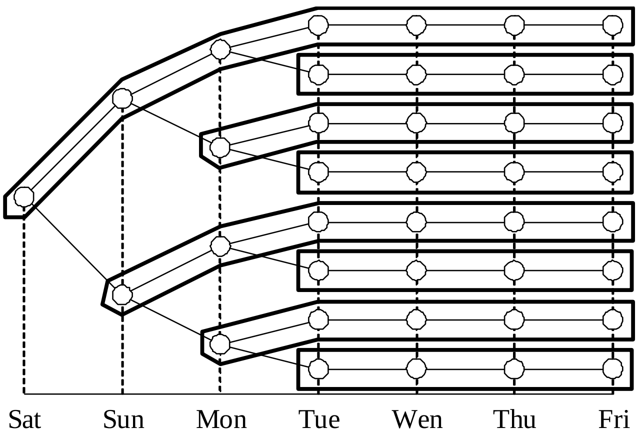

Figure 2.Partitioning approach for the market unit commitment problem

The reader may appreciate that this partitioning scheme fits into the standard input format for stochastic programming problems (Birge, Dempster et al. 1987).

The algorithm proposed in this paper has been applied to this variety of problems. Input data are introduced in SMPS format. CPLEX 7.5 (ILOG 2003) was used as the LP solver as well as the MIP solver for the individual subproblems resolutions. Problems have been run in a personal computer with 1 GB of RAM memory and a CPU of 3 GHz.

Name	Number of Scenarios	Lower Bound	Solution	Iterations	CPU seconds
Ex1	4	-57.5	-56.25	6	0.23
Ex1	9	-60.68	-58.92	7	0.96
Ex1	36	-62.87	-61.22	9	3.35
Ex1	121	-64.15	-62.29	9	5.82
Ex1	441	-63.11	-61.31	10	27.65

Table 2. Computational results for the Ex1 problem.

Name	Number of Scenarios	Lower Bound	Solution	Iterations	CPU seconds
Ex2	4	-57.5	-56.25	6	0.25
Ex2	9	-60.68	-58.92	7	0.98
Ex2	36	-62.87	-60.86	10	4.81
Ex2	121	-64.14	-62.06	10	7.04
Ex2	441	-64.27	-60.21	10	38.54

Table 3. Computational results for the Ex2 problem.

Name	Number of Scenarios	Lower Bound	Solution	Iterations	CPU seconds
Ex3	4	-54.05	-50.60	8	0.87
Ex3	9	-57.14	-54.54	6	1.31
Ex3	36	-59.30	-57.38	8	3.46
Ex3	121	-59.82	-56.54	9	12.92
Ex3	441	-60.60	-55.62	10	79.84

Table 4. Computational results for the Ex3 problem.

Name	Number of Scenarios	Lower Bound	Solution	Iterations	CPU seconds
Sslp_5_25_50	50	-121.6	-121.6	26	9.50
Sslp_5_25_100	100	-127.37	-127.37	24	26.09
Sslp_15_45_5	5	-262.4	-262.4	31	76.75
Sslp_15_45_10	10	-260.5	-260.5	35	1180
Sslp_15_45_15	15	-253.6	-253.601	36	2870
Sslp_10_50_50	50	-364.62	-364.62	149	5162
Sslp_10_50_100	100	-354.19	-354.19	172	25565
Sslp_10_50_500	500	---	---	---	>30000

Table 5. Computational results for the Sslp problem.

Name	Number of Scenarios	Lower Bound	Solution	Iterations	CPU seconds
Sizes	3	225441	226499	8	4.75
Sizes	5	224586	225984	11	16.75
Sizes	10	223355	225226	10	42.15

Table 6. Computational results for the Sizes problem.

Name	Number of Scenarios	Lower Bound	Solution	Iterations	CPU seconds
Dcap233_200	200	1833.47	1864.50	30	358
Dcap233_300	300	1642.93	1671.22	26	621
Dcap233_500	500	1736.70	1781.49	22	760
Dcap342_200	200	1618.12	1733.28	41	873
Dcap342_300	300	2065.67	2143.35	35	1154
Dcap342_500	500	1903.02	2013.00	38	3057
Dcap243_200	200	2321.22	2344.16	32	464
Dcap243_300	300	2556.96	2594.64	34	919
Dcap243_500	500	2165.51	2186.42	42	2786
Dcap332_200	200	1059.01	1129.34	46	1013
Dcap332_300	300	1251.09	1366.14	46	1355
Dcap332_500	500	1587.13	1669.16	42	4822

Table 7. Computational results for the Dcap problem.

Name	Number of Scenarios	Upper Bound	Solution	Iterations	CPU seconds
Muc	8	5.404626	5.401882	13	3684

Table 8. Computational results for the Muc problem.

The obtained numerical results may be analyzed from different points of view. First of all, the simplicity of the examples *Ex1*, *Ex2* and *Ex3*, shows the limitations of using the exact convexification of the recourse functions of independent scenario subproblems in the algorithm based on decomposition techniques. However, the method seems to achieve the optimal solution when the number of scenarios increases. The CPU time is similar to the time presented in (Ahmed, Tawarmalani et al. 2004).

This need of incorporating branch and bound techniques is clear when observing the results of the *Dcap* problem. In most of these problems there is a gap between the solution and the lower bound provided by the algorithm. An issue to take into account appears when comparing these numerical results with the solutions of the *Dcap* problem given in (Ahmed and Garcia 2002). Although the solutions given by our method are far away from those solutions, the lower bounds are really close to them, a feature that needs to be considered in future enhancements of the method.

The problem *Sslp* seems to behave in a different way to the remaining examples. For the problem instances solved, the lower bound coincides with the solution obtained. This suggests that for this particular example the convexification of the sum of recourse functions may be equal to the sum of convexifications. These problems are solved in (Sen and Higle 2004) by the convexification of the feasible regions of the individual scenario subproblems, which leads to obtaining the lower convex envelope of the recourse functions. It should be interesting to test the behavior of their algorithm, named *C3D2* over the numerical examples of the *Dcap* problems. In any case, the computation times obtained by the *C3D2* algorithm are really better than those achieved with our method. In fact, our method is not able to solve instances with more that 500 scenarios.

The *Sizes* problems as well as the *Muc* problems are examples of models that contain mixed-integer variables at any stage. The solution of our method applied to the *Sizes* problem does not improve the solutions in (Ahmed, Tawarmalani et al. 2004). However, the computation time really outperforms that of his method. The *Muc* problem is the biggest problem of all the tested ones. The numerical results of the application of the method to this problem are extraordinary if we take into account that CPLEX 7.5 was not able to solve the problem within a limit of 24 hours. In this case, we only had to execute two phases to reach an accuracy of less than 0.1 %. A note on the application of the method to UC commitment problems needs to be outlined. The large dimension of the inner subproblem of the max-min procedure of (12) suggests that an efficient application of the method should avoid the optimization of the dual problem (D). This suggestion is confirmed by the numerical results of table 8. In this case, the algorithm was stopped after the phase 2 of the method was done.

4 CONCLUSION

This paper has presented an extension of Benders' decomposition algorithm to face the solution of multistage problems with integer variables at any stage. The extension is based on the idea of approximating the lower convex envelope of the recourse functions of the individual scenario subproblem. The convexification presented in this paper considers the implicit domain of the recourse function imposed by the collection of tender variables. Additionally, the paper suggests the use of the algorithm in a sequential way, improving the approximation of the recourse function by calculating computationally cheaper cuts prior to more expensive cuts.

We have illustrated the importance of the proposed approach by means of its application to a battery of academic examples as well as to a real-size weekly stochastic market unit commitment problem. The results obtained by our method clearly indicate that the algorithm has to be considered as an adequate alternative to the direct resolution of real sized problems as UC problems. The numerical results also indicate the limitations of considering the convexification of the recourse function in the development of decomposition-based algorithms for stochastic integer problems. The computational time is high in some examples, in particular for those problems with a large number of scenarios. This may be caused by the use of a multicut strategy that creates one cut in the master problem for each subproblem solved in each iteration. The use of a monocut version as well as the combination of the method with branch and bounds techniques appears as subjects of further study.

5 REFERENCES

Ahmed, S. and R. Garcia (2002). The DCAP Test Problems. http://www.isye.gatech.edu/~sahmed/siplib/dcap/dcap.html#[1].

Ahmed, S., M. Tawarmalani, et al. (2004). "A finite branch-and-bound algorithm for two-stage stochastic integer programs." Mathematical Programming **100**: 355-377.

Benders, J. F. (1962). Partitioning Procedures for Solving Mixed Variables Programming Problems. *Numerische Mathematik, 4*, 238-252.

Birge, J. R., M. A. H. Dempster, et al. (1987). "A standard input format for multiperiod stochastic linear programs." Mathematical Programming Society Committee on Algorithms Newsletter **17**: 1-19.

Birge, J. R., & Louveaux, F. (1988). A Multicut Algorithm fo Two-Stage Stochastic Linear Programs. *European Journal of Operational Research, 34*.

Birge, J. R., & Louveaux, F. (1997). *Introduction to Stochastic Programming* (1st ed.). New York: Springer.

Carøe, C. C., & Tind, J. (1998). L-Shaped Decomposition of Two-Stage Stochastic Programs with Integer Recourse. *Mathematical Programming, 83*.

Cerisola, S., Á. Baíllo, et al. (2005). "Stochastic Power Generation Unit Commitment in Electricity Markets: A Novel Formulation and a Comparison of Solution Methods." Submitted to Operations Research.

Dupacová, J., Consigli, G., & Wallace, S. W. (2000). Scenarios for Multistage Stochastic Programs. *Annals of Operations Research, 100*, 25-53.

Geoffrion, A. M. (1972). Generalized Benders Decomposition. *Journal of Optimization theory Applications (JOTA), 10*, 237-259.

Geoffrion, A. M. (1974). Lagrangean relaxation for integer programming. *Mathematical Programming Study, 2*, 82-114.

ILOG. (2003). http://www.ilog.com.

Laporte, G., & Louveaux, F. (1993). The Integer L-Shaped Method for Stochastic Integer Programs with Complete Reourse. *Operations Research Letters, 13*.

Morton, D. (1993). *Algorithmic Advances in Stochastic Programming* (SOL 93-6): Systems Optimization Laboratory, Department of Operations Research, Stanford University.

Olaf E. Flippo, A. H. G. R. K. (1993). Decomposition in general mathematical programming. *Mathematical Programming, 60*, 361-382.

Rockafellar, R. T., & Wets, R. J.-B. (1991). Scenarios and Policy Aggregation in Optmization Under Uncertainty. *Mathematics of Operations Research, 16*.

Ruszczynski, A. (1997). Decomposition methods in stochastic programming. *Mathematical Programming, 79*, 333-353.

Sen, S. and J. L. Higle (2004). "The C3 Theorem and a D2 Algorithm for Large Scale Stochastic Mixed-Integer Programming: Set Convexification." Mathematical Programming.

Sen, S. and D. P. Kothari (1998). "Optimal Thermal Generating Unit Commitment: A Review." Electrical Power & Energy Systems **20**(7): 443-451.

Takriti, S., Krasenbrink, B., & Wu, L. S.-Y.-. (2000). Incorporating Fuel constraints and Electricity Spot Prices into the Unit Commitment Problem. *Operations Research, 48*(2), 268-280.

Van der Vlerk, M. (1995). *Stochastic programming with integer recourse.* University of Groningen.

Van Slyke, R. M., & Wets, R. (1969). L-Shaped Linear Progams with Applications to Optimal Control and Stochastic Programming. *SIAM Journal on Applied Mathematics, 17*(4), 638-663.