SEQUENTIAL CUT REFINEMENT METHOD IN MULTISTAGE STOCHASTIC INTEGER PROGRAMMING: APPLICATION TO A UNIT COMMITMENT PROBLEM

SANTIAGO CERISOLA, ÁLVARO BAÍLLO, ANDRÉS RAMOS

Escuela Técnica Superior de Ingeniería ICAI, Universidad Pontificia Comillas,

Alberto Aguilera 23, 28015 Madrid, Spain, santiago.cerisola@iit.upco.es

RALF GOLLMER

Institut für Mathematik University Duisburg-Essen

gollmer@mail.math.uni-duisburg.de

Subject classification:

- Programming: Stochastic programming, Integer programming, Benders decomposition.
- Production/scheduling: unit commitment decisions under uncertainty.

Area of Review: Environment, Energy, and Natural Resources.

Benders' method usually divides the collection of decision variables of a two-stage mathematical problem into two sets: a first set that comprises the collection of variables that represent first-stage decisions and a second set that includes the collection of variables that represent recourse actions or second-stage decisions. Usually, integer variables appear in the first set whereas the second set is formed by continuous variables with the purpose of having a convex recourse function. Convexity is then exploited in the construction of algorithms to obtain an optimal solution. In this paper we present an extension of Benders' method to deal with multistage problems with integer variables in any stage. This extension is based on the idea of sequentially refining the linear cuts that are used to represent an approximation of the recourse function in the master problem. Computationally cheaper cuts are obtained first and more expensive ones are calculated only if the desired tolerance is not reached. The paper includes an application of the proposed method to a stochastic weekly power generation unit commitment model.

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 scenario sampling and simulation including variance reduction 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 manner of representing uncertainty in this type of problems when probability distributions are approximated with discrete functions is through a scenario tree (Dupacová, Consigli and Wallace (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 situations with two decision stages, Benders' decomposition technique is the most extended one (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 an extension to nested situations 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 at each iteration by means of a disjunctive logic constraint: only one of these cuts is allowed to be active in subsequent iterations. Other authors such as Flippo and Kan (1993) use generalized duality instead of linear duality with the purpose of approximating the recourse function. Carøe and Tind (1998) generate a non-continuous approximation of the recourse function by using subbadditive theory in the solution of the second-stage problem. However, their technique generates cuts that are neither convex nor continuous, which complicates its use with algorithmic purposes.

Convexification is a natural approach to address the solution of these problems. Generalized Benders' decomposition (GBD), originally proposed by Geoffrion, A. M. (1972), uses nonlinear duality to approximate the convexification of the recourse function. An extension of GBD is the sequential cut refinement method presented in this paper.

Alternative to L-shaped methods, that exploit 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, Arthut M. (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)).

The vast majority of these methods are not computationally efficient and frequently require heuristics to improve their performance. In this paper we suggest a sequential approach to execute Benders' algorithm by generating 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.

We include an application of this algorithm to solve a stochastic unit commitment (UC) problem. In Takriti and Birge (2000), the traditional Lagrangean relaxation of the demand constraint is complemented with the use of mixed integer programming techniques in order to obtain a final feasible schedule. In Nowak and Römisch (2000), Lagrangean relaxation is used to relax the non-anticipativity constraints that link the different scenarios of a stochastic UC problem. In the work presented in this paper, the weekly stochastic UC problem is decomposed into individual subproblems each one representing one day of operation. This decomposition perfectly fits within our

extension of Benders' method. The binary (or integer) commitment decisions for thermal units are the source of nonconvexities in the resulting recourse functions.

The rest of the paper is organized as follows. Section 2 briefly reviews Benders' decomposition or L-shaped meted, GBD and the extension of GBD that is used here. Section 3 presents the sequential cut refinement method as well as its natural extension to multistage problems via nested decomposition. We include an academic example to illustrate the method. Section 4 tests the efficiency of the proposed method over different stochastic instances of a particular UC problem. Finally, section 5 presents the main conclusions of this research.

2 L-SHAPED DECOMPOSITION METHODS

2.1 The L-Shaped Method

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

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

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):

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

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 $\left\{x / 0 \geq \pi^i(h-Tx)+\rho^i a_2\right\}_{i\in I^*}$, where $\left\{(\pi^i,\rho^i)_{i\in I^*}\right\}$ 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^i and θ^i be the optimal dual value and optimal solution of a feasible subproblem (SP_{x^i}) when a certain first-stage solution x^i has been proposed. Then, the following is a lower bound for the recourse function:

$$\begin{aligned} Q(x) &\geq \pi^i(h-Tx)+\rho^i a_2 = \pi^i(h-Tx+Tx^i-Tx^i)+\rho^i a_2 = \\ &= \pi^i(h-Tx^i)+\rho^i a_2+\pi^i(-Tx+Tx^i) = \theta^i+\pi^i T(x^i-x) \end{aligned} \qquad (4)$$

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

$$
(RMP) \qquad \begin{aligned} & \min cx + \theta \\ & 0 \geq \theta^i + \pi^i T(x^i - x) \quad i \in \tilde{I}* \\ & \theta \geq \theta^i + \pi^i T(x^i - x) \quad i \in \tilde{I} \\ & x \in X \end{aligned} \tag{5}
$$

where $\tilde{I}$ is a subset of I, $\tilde{I}*$ is a subset of $I*$ and x^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^i. This first-stage solution is then used to evaluate the recourse function by solving the corresponding subproblem (SP_{x^i}). The description of the recourse function in (RMP) is enhanced with an optimality cut in case of subproblem feasibility. In the other case, the feasibility region of (RMP) is constrained with a feasibility cut. 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 $i = 0$. Set $\theta \equiv 0$ at the initial iteration
Step 1	Solve (RMP) and obtain solution x^i and lower bound $\underline{z} = v(RMP)$
Step 2	Solve (SP_{x^i})
	If (SP_{x^i}) is infeasible let $i \in \tilde{I}*$, and obtain π^i
	If (SP_{x^i}) is feasible set $i \in \tilde{I}$, obtain π^i and compute upper bound $\bar{z} = cx^i + v(SP_{x^i})$
Step 3	(stopping rule)
	If $\left(\bar{z} - \underline{z}\right)/\bar{z} < tol$ stop, x^i is the optimal solution, else go to Step 1

Algorithm 1. L-Shaped method.

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

As commented before, this method was originally conceived to address two-stage problems with a linear second stage. The introduction of integrality requirements for the variables of the subproblem significantly complicates the application of the previous approach, given that the recourse function turns out to be non-convex. If the L-shaped method is applied to such a two-stage problem by simply relaxing these integrality requirements the approximation that is obtained for the recourse function may be too inaccurate to guide the relaxed master problem toward the solution of the original problem.

2.2 Generalized Benders Decomposition

The generalized Benders' decomposition algorithm (Geoffrion, A. M. (1972; Holmberg (1994)) consists of iterating between the relaxed master problem and the Lagrangean Relaxation of the subproblem, where the relaxed equations are those that connect both stages. The solution of the subproblem via Lagrangean Relaxation (LR) has the effect of convexifying the recourse function in case the subproblem incorporates integrality requirements. This convexification considers the domain of the recourse function to be the complete Euclidean space. Given a first-stage solution, x^i, the Lagrangean dual of subproblem (SP_{x^i}) is solved

$$(PR_{x^i}) \qquad \theta^i = \max_{\lambda} \min_{y} \left\{ qy + \lambda(Tx^i + Wy - h), y \in Y \right\} \tag{6}$$

where the inner minimization problem in (PR_{x^i}) is known as the Lagrangean subproblem. The optimal solution λ^i and optimal value θ^i, are used to build up the cut that outer approximates the recourse function in (RMP)

$$\theta \geq \theta^i - \lambda^i T(x^i - x) \tag{7}$$

Notice that λ^i can be interpreted as the opposite of the dual variable obtained for the linear situation.

The generalized Benders' decomposition treats infeasibility situations in a similar manner as the traditional algorithm does. For the MIP situation, the solution of (6) is preceded by the solution of the subproblem that minimizes the sum of infeasibilities. Noticing that infeasibilities can only be caused by the coupling constraints, the minimization of infeasibilities subproblem for a certain first-stage solution, x^i, can be formulated as:

$$Q^*(x^i) = \min_{y,\delta^+,\delta^-} \left\{ \delta^+ + \delta^-, Wy + \delta^+ - \delta^- = h - Tx^i, y \in Y, \delta^+, \delta^- \geq 0 \right\} \qquad (8)$$

The method exploits the fact that it is equivalent to identify that the subproblem is infeasible ($Q^*(x^i) > 0$), or that the convexification of Q^* for the first-stage proposal x^i has a positive value. For this reason, and in order to obtain an infeasibility cut for non-valid solutions, the direct solution of (8) is replaced by solving its LR formulation:

$$\theta^i = \max_{\lambda} \min_{y,\delta^+,\delta^-} \left\{ \delta^+ + \delta^- + \lambda(\delta^+ - \delta^- + Tx^i + Wy - h), y \in Y, \delta^+, \delta^- \geq 0 \right\} \qquad (9)$$

It is immediate that the above problem is equivalent to

$$(PR_{x^i})^* \qquad \theta^i = \max_{\lambda \in [-1,1]} \min_{y} \left\{ \lambda(Tx^i + Wy - h), y \in Y \right\} \qquad (10)$$

an expression that recovers the necessary condition for feasible solutions in the generalized Benders' algorithm (Geoffrion, A. M. (1972)). To summarize, the GBD method iterates between the relaxed master problem formulated in (5) and the pair of problems $(PR_{x^i})^*$ and (PR_{x^i}). The algorithm proceeds as the traditional algorithm does. The extension of this algorithm to a nested situation is immediate.

2.3 Generalized Benders' Decomposition Extension

The extension of the generalized Benders decomposition presented in this section is based on a geometrical interpretation of the LR method proposed by Lemarechal and Renaud (2001). According to this extension, the recourse function $Q(x) = \min\{qy, Wy = h - Tx, y \in Y\}$ may be seen as the lower envelope of the epigraph associated to the problem, $\text{epi}\, G$

$$\text{epi}\, G = \{(r, r_0) / \exists y \in Y \text{ with } r = Wy - h, r_0 \geq qy\} \tag{11}$$

It is important to notice that this expression of the epigraph does not take into account the fact that the values that r can take are restricted to the set $\{-Tx, x \in X\}$. Based on the preceding expression of the epigraph, the Lagrangean dual of problem (SP_{x^i}) can be alternatively formulated as follows

$$(PR_{x^i}) \qquad \max_{\lambda} \left\{ \min_{y, r_0, r} \{r_0 + \lambda r, r_0 \geq qy, r = Wy - h, y \in Y\} + \lambda T x^i \right\} \tag{12}$$

Where r_0 and r simply are two auxiliary variables. It is evident that the solution of (12) satisfies $r_0 = qy$. An interesting geometric interpretation of problem (PR_{x^i}) is that it provides the convexification of $\text{epi}\, G$. However, as indicated before, this convexification does not take into account the fact that the values that r can take are restricted to the set $\{-Tx, x \in X\}$. Due to this, the inner minimization in (12) has a solution with smaller objective value than desired. This implies that the convexification derived from (PR_{x^i}) provides an approximation for the recourse function that runs below the optimal one.

Let us consider now the possibility of restricting the feasibility region of r. As mentioned, the feasibility region for r depends on the range of values that x can take.

This establishes a stronger link between the master problem and the Lagrangean dual of the subproblem and transforms r into a perturbation variable. The inclusion of this feasibility set yields the following perturbation problem:

$$(PPR_{x^i})\ \max_{\lambda}\left\{\min_{x,y,r_0,r}\left\{r_0+\lambda r, r_0 \geq qy, r=Wy-h, y\in Y, r\in\{-Tx, x\in X\}\right\}+\lambda Tx^i\right\} \quad (13)$$

Given the feasibility region defined for r, it is clear that $r=-Tx$ for some $x\in X$. Additionally, the solution of (PPR_{x^i}) still satisfies $r_0=qy$. Hence, the following equivalent formulation can be considered:

$$(PPR_{x^i}) \qquad \max_{\lambda}\left\{\min_{x,y}\left\{qy-\lambda Tx, Tx+Wy=h, x\in X, y\in Y\right\}+\lambda Tx^i\right\} \quad (14)$$

Prior to solving problem (PPR_{x^i}), it is necessary to check its feasibility for the proposed value x^i by means of the following auxiliary problem.

$$(PPR_{x^i})^* \qquad \max_{\lambda\in[-1,1]}\left\{\min_{x,y}\left\{-\lambda Tx, Tx+Wy=h, x\in X, y\in Y\right\}+\lambda Tx^i\right\} \quad (15)$$

The proposed perturbation problem improves the approximation of the lower convex envelope of the recourse function compared to the one the LR method of the GBD algorithm obtains.

The solution of problem (14) may be considered inefficient. On the one hand, it seems to have the same number of variables than the original problem, (1). On the other hand, it is embedded in an iterative algorithm that requires its solution an indefinite number of times. However, in practical applications, the collection of first-stage variables that perturb the right hand side of the second-stage problem is usually a small subset of the set of first-stage variables, which reduces the size of (14) in comparison to that of (1). The proposed method is also adequate for nested situations.

3 SEQUENTIAL CUT REFINEMENT METHOD

3.1 A description of the method

The sequential method that we propose in this section 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 five phases where each phase is characterized by the way of computing the Benders' cuts.

3.1.1 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 passes we solve the linear relaxation of the (RMP) 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.

3.1.2 Phase 2

In phase 2 we reincorporate the integrality requirements that we removed in phase 1. When we traverse the chain of problems forward, we solve their MIP version, (RMP) (this also holds for phases 3 to 5). In contrast, when we traverse the chain of problems backward, we relax integrality requirements. Hence, Benders' cuts in this second phase also correspond to those of the traditional Benders' decomposition

algorithm. The difference with phase 1 lies in that the cuts for each subproblem are calculated for points that satisfy the integrality requirements of its ancestor. The cost of this improvement is the effort required to solve the MIP version of the subproblems in the forward pass.

In the nested Benders' decomposition algorithm for linear problems, the forward pass stops as soon as an infeasible subproblem is found (there is no sense in proposing values that come from previous stages with infeasible solutions). When passing from phase 1 to phase 2, a subproblem may turn out to be infeasible for two reasons. On the one hand, the linear relaxation of the subproblem may be infeasible. In that case its MIP version is not solved and an infeasibility cut is generated as in phase 1. On the other hand, the subproblem may be linear feasible but MIP infeasible. In that case a feasibility cut must be generated as in the GBD method, via the solution of $(PR_{x^i})^*$.

In each iteration, a lower bound is given by the solution of the master problem and an upper bound is determined by the evaluation of the objective function of the complete problem for the current solution. Observe that as long as the method to approximate the recourse function does not produce the exact convexification, the lower bound may never reach the upper bound. Phase 2 finishes when the difference of the primal values obtained in two consecutive iterations is lower than a specified tolerance.

3.1.3 Phase 3

This phase presents a refinement of the method for computing linear cuts with respect to previous phases. In backward passes, we first solve the subproblems with the integrality requirements relaxed. Then we take the dual variables corresponding to the coupling constraints and we evaluate the objective function of the Lagrangean dual of the subproblem. To do so we solve the Lagrangean subproblem (the Lagrangean

subproblem is the inner minimization of problem (PPR_{x^i})) for that value of the multipliers. A more accurate linear cut is obtained because the value of the objective function of the lagrangean dual of the subproblem for those multipliers will be greater or equal than the objective function of the subproblem with its integrality requirements relaxed.

In a two-stage situation, 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.

This phase finishes when the difference of the primal values obtained in two consecutive iterations is lower than a specified tolerance.

3.1.4 Phase 4

The cuts calculated in this phase are certainly harder to compute than those of phase 3 although not necessarily better. In backward passes, the values of the dual variables selected to evaluate the Lagrangean dual objective function are the ones obtained from the solution of the MIP version of the subproblem. More precisely, these dual variables are obtained solving the LP problem that results when integer variables are fixed to the optimal values achieved so far for the MIP problem (as in a branch-and-bound method).

The computation of these cuts is suggested by the typical shape of one-dimensional recourse functions (Wets (1996)) and by the necessity of constraining the computation of the convexification to the domain of the recourse function imposed by the first-stage variables.

The computational cost of computing this cut is higher than the cost of the cut presented in phase 3 because it implies the solution of two MIP problems and the evaluation of the Lagrangean dual objective function, while the cut of phase 3 requires the solution of one LP, one MIP, and the evaluation of the Lagrangean dual objective function.

3.1.5 Phase 5

In this last phase the extension of the GBD method that we have presented in the previous section must be applied. This requires a significant computational effort.

3.1.6 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 an 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 cut refinement method may be stopped at any phase if the tolerance required for the solution is reached. This is useful because it permits avoiding phases 4 or 5, which are radically more time consuming than the previous ones.

A major drawback of the sequential cut refinement 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 execution of problem $(PPR_{x^i})^*$, 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
Phase 3	MIP	LP + MIP Lagrangean Subproblem
Phase 4	MIP	MIP + MIP Lagrangean Subproblem
Phase 5	MIP	Max-Min Subproblem

Table 1. Cut refinement method.

3.2 Example

We now present a small academic example to illustrate the performance of the cut refinement method. Consider the problem

$$\begin{aligned} &\min -0.3x - 1.5y - z \\ &0 \le x \le 5 \\ &x + y \le 3.7 \\ &y + z \le 5.2 \\ &y \ge 0, z \ge 0 \\ &y \in \mathbb{Z}, z \in \mathbb{Z} \end{aligned}$$

with optimal solution -6.71 reached at $x = 0.7$, $y = 3$, $z = 2$.

In order to solve this program by Benders decomposition we formulate the following master problem

$$\begin{aligned} &\min -0.3x + Q(x) \\ &0 \le x \le 5 \end{aligned}$$

and the subproblem

$$\begin{aligned} &Q(x) = \min -1.5y - z \\ &y \le 3.7 - x \\ &y + z \le 5.2 \\ &y \ge 0, z \ge 0 \\ &y \in \mathbb{Z}, z \in \mathbb{Z} \end{aligned}$$

The recourse function $Q(x)$ is depicted in the next figure together with the objective function of the master problem.

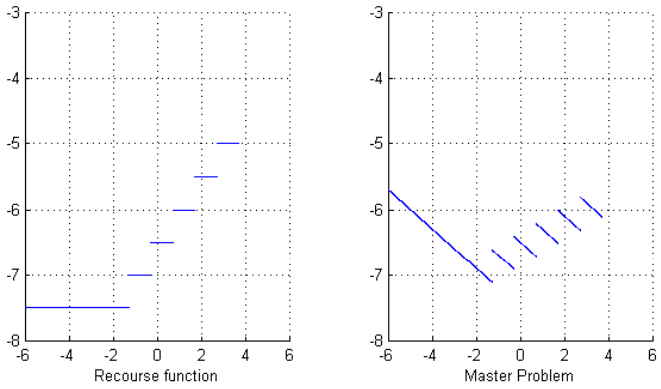

Figure 1. Recourse function and objective function of the master problem.

Let us now solve this problem by using the sequential cut refinement method. Phase 1, that solves the linear relaxation of above problem, finishes with an optimal solution given by $x = 0$, $y = 3.7$, $z = 1.5$. The relaxed master problem provides the following approximation for the complete problem when phase 1 finishes:

$$\begin{aligned} &\min -0.3x + \theta \\ &0 \le x \le 5 \\ &x \le 3.7 \\ &\theta \ge 0.5x - 7.05 \end{aligned}$$

In phase 2 we solve the MIP version of the master problem. However, as this example considers a continuous first-stage variable the value proposed by the master problem is again $x = 0$. Additionally, the subproblem is integer feasible for this proposal with solution $y = 3$, $z = 2$. Hence, no iterations of phase 2 are needed.

In phase 3, the linear relaxation of the subproblem is solved for the last value proposed by the master problem ($x = 0$).

$$\begin{aligned} &Q(0) = \min -1.5y - z \\ &y \le 3.7 \quad : \quad \pi \\ &y + z \le 5.2 \\ &y \ge 0, z \ge 0 \end{aligned}$$

The solution of this problem yields $\pi = -0.5$ for the dual variable of the coupling constraint. As a consequence, Phase 3 continues with the solution of the Lagrangean subproblem for $\lambda = 0.5$

$$\begin{aligned}
&\theta = \min -1.5y - z - 0.5x \\
&x + y \le 3.7 \\
&y + z \le 5.2 \\
&0 \le x \le 5 \\
&y \ge 0, z \ge 0 \\
&y \in \mathbb{Z}, z \in \mathbb{Z}
\end{aligned}$$

The solution of this Lagrangean subproblem is $\theta = -6.85$, which leads to an approximation of the recourse function at $x = 0$ given by $\theta + 0.5Tx^i = -6.85 + 0.5 \times 0 = -6.85$. With this, the Benders cut takes the following form

$$\theta \ge 0.5x - 6.85$$

This cut improves the approximation of the recourse function provided at the end of phase 1. The solution of the master problem yields the same first-stage proposal, $x = 0$. Hence, the stopping criterion for phase 3 is satisfied and the algorithm switches to phase 4.

In phase 4, we solve the MIP version of the subproblem to obtain a new multiplier value, $\pi = 0$. The Lagrangean subproblem for $\lambda = 0$ is given by

$$\begin{aligned}
&\theta = \min -1.5y - z \\
&x + y \le 3.7 \\
&y + z \le 5.2 \\
&0 \le x \le 5 \\
&y \ge 0, z \ge 0 \\
&y \in \mathbb{Z}, z \in \mathbb{Z}
\end{aligned}$$

The solution for this Lagrangean subproblem is $\theta = -6.5$. The new cut has now the form $\theta \ge -6.5$. This new cut improves the outer approximation provided by phases 1 to 3. Due to this, the new solution of the master problem gives a different proposal,

$x = 0.7$. The solution of the subproblem for this proposal is $y = 3$, $z = 2$. The final stopping criterion is satisfied because the lower bound (solution of the master problem) and the upper bound (evaluation of the objective function of the complete problem) coincide.

In the next figures we depict the evolution of the recourse function approximation that the sequential cut refinement method produces. The figures of the left column represent the approximation of the recourse function and the figures of the right column show the approximation provided by the relaxed master problem. The figures of the first row represent the approximation provided at the end of phase 1. Those of the second row depict the approximation provided at the end of phase 3. The final row depicts the approximation at the end of phase 4.

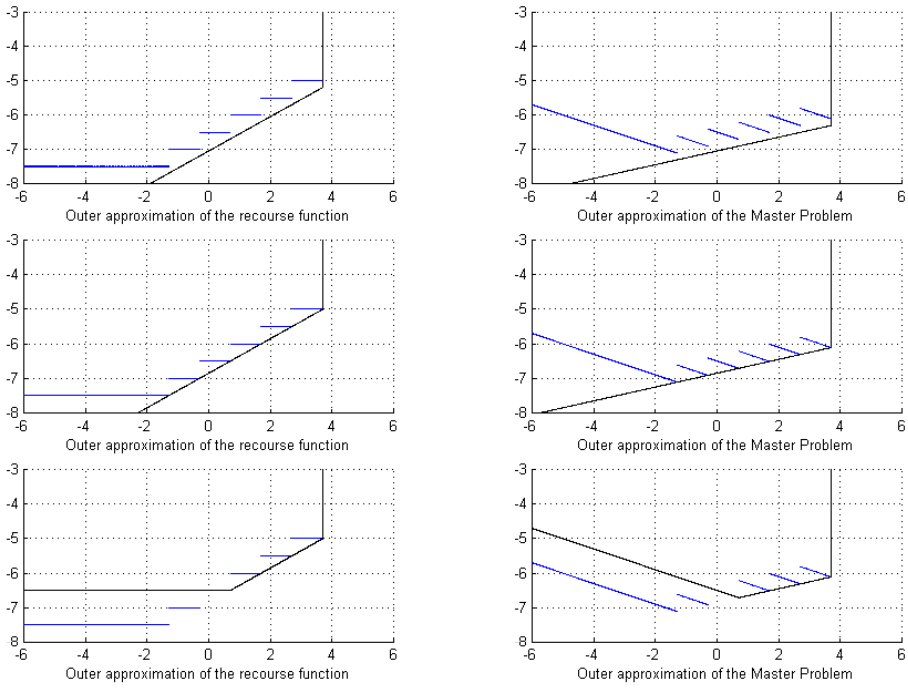

Figure 2. Outer approximation of the recourse function and of the objective function of the master problem.

4 APPLICATION TO A POWER GENERATION UNIT COMMITMENT PROBLEM

The power generation unit commitment (UC) problem consists of determining the optimal commitment schedule of a system of generation units in order to cover the demand for power of a region during a certain time horizon (e.g. one week). Given that power generation must meet the demand for power at every moment, the UC problem must take into account not only the commitment status of the generation units, but also the power they will provide in each time interval (e.g. one hour).

A generation system typically consists of different type of units, including nuclear plants, fossil-fueled thermal units (e.g. coal-fired units or combined-cycle gas turbines CCGTs), hydro plants and others. These types of units present differences both in their cost structure and in their operation constraints that must be considered when deciding a generation schedule. In general, the UC problem is oriented to minimize the cost of power production while meeting the demand for power at every load level and satisfying the operation constraints of the committed units.

The UC problem is a MIP problem due to the presence of the binary or integer variables that represent the commitment status of the generation units. A variety of approaches have been adopted to solve the UC problem (Sheblé (1994)). In recent years, authors have focused on the development of new solution methods and on the introduction of new modeling features (Hobbs *et al.* (2001)).

The stochastic UC problem has also received much attention. The main source of uncertainty that authors have considered is the demand for power (Nowak and Römisch (2000)). The increasing importance of wind power, together with its inherent stochastic nature, can be seen as another contribution to the uncertainty of the demand for

conventional power generation. A different line of research has been oriented to represent uncertainty in the revenues obtained by generators for their power production in the new deregulated environment (Valenzuela and Mazumdar (2003)).

Even with modern commercial mixed-integer programming codes the direct solution of stochastic UC problems of realistic size would consume excessively much time and computer memory. Hence, decomposition techniques have traditionally been used for this purpose. Two main approaches are typically adopted. The first one is scenario decomposition, which splits the stochastic problem into as many deterministic problems as scenarios are being considered by relaxing non-anticipativity constraints (Takriti and Birge (2000), Carøe and Schultz (1998)). The alternative is to relax the constraints that link the operation of the generation units (e.g. the demand constraint, which forces the sum of the power output of the different units to meet the demand for power at each load level). This yields a collection of stochastic subproblems, one for each generation unit (Carpentier *et al.* (1996), Dentcheva and Römisch (1998), Takriti, Krasenbrink and Wu (2000)).

The decomposition approach that we propose yields one subproblem for each node of the scenario tree and coordinates the solution of these subproblems by means of the method described in previous sections. This section presents the application of our approach to the solution of a realistic UC problem.

The particular problem that we have solved is the UC problem of the German utility Vereinigte Energiewerke AG (VEAG). Its total capacity is about 13,000 megawatts (MW) including a hydro capacity of 1,700 MW; the system peak loads are about 8,600 MW. The particular problem of the VEAG system consists of 168 periods, representing a weekly horizon, 35 thermal units and 22 pumped-storage units. Pumped-storage units pump water from their lower reservoir to their upper reservoir during low-

demand hours in order to prevent thermal units from stopping and losing the thermal energy accumulated in their boilers. Pumped-storage units release this water during high-demand hours, thus reducing the need for thermal power.

The matrix staircase structure of this problem for a one-scenario case has been represented in Figure 3, where each block corresponds to one day.

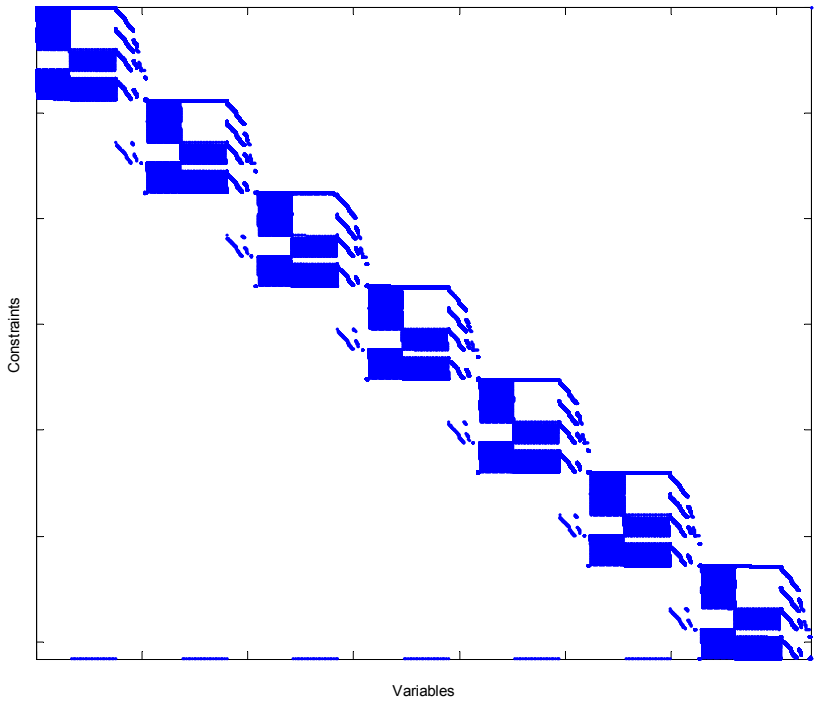

Figure 3. Constraints matrix for the deterministic seven-day UC problem.

We have coded the decomposition algorithm, including the sequential cut refinement method, in Concert Technology 1.2, an optimization library that connects with the optimizer CPLEX 7.5 (ILOG (2003)).

4.1 Problem formulation

We present a simplified formulation of the UC problem because the purpose of this paper is not to propose a novel formulation or a new modeling feature. In particular, we omit the representation of uncertainty in our formulation to avoid the complexity

this would incur. One way to do this would be to represent the operation of the generation system in each possible scenario and explicitly formulate non-anticipativity constraints. An alternative would be to use a formulation based on the nodes of the scenario tree that establishes links between each node and its ancestor.

The objective of the stochastic UC problem is to minimize the expected short-term operational cost of the generation system

$$\min \sum_{t\in T}\sum_{i\in I}\left(C_i c_{it} + S_i s_{it}\right) \tag{16}$$

where $t \in T$ represent time intervals (hours), $i \in I$ refer to thermal units and $j \in J$ correspond to hydro units. The short-term cost of the generation system only includes the cost due to fuel consumption in thermal plants: C_i is the fuel cost of thermal unit i per unit of fuel consumed, in €/Tcal, c_{it} is the fuel consumed by thermal unit i during time interval t, in Tcal, S_i is the cost of the fuel consumed during the startup of unit i, in €, and s_{it} represents the start-up decision for thermal unit i at the beginning of time interval t (0/1).

The net power output of the generation system must meet the demand for power in each time interval t:

$$\sum_{i\in I} p_{it} + \sum_{j\in J}(p_{jt} - w_{jt}) = D_t \tag{17}$$

where p_{it} is the power output of thermal unit i, in MW, p_{jt} is the power produced by hydro unit j, in MW, w_{jt} is the power consumed by hydro unit j due to pumped storage, in MW, and D_t is the demand for power in time interval t, in MW.

To prevent the collapse of the system in case of the outage of a generation unit, a certain level of power reserve is typically required in each time interval t:

$$\sum_{i \in I} u_{it} (p_i^{\max} - p_{it}) + \sum_{j \in J} (p_j^{\max} - p_{jt}) \geq R_t \tag{18}$$

where u_{it} is the commitment status of thermal unit i (0/1), $p_i^{\max}$ is the net maximum power output of thermal unit i, in MW, $p_j^{\max}$ is the net maximum power output of hydro unit j, in MW, and R_t is the level of power reserve required in time interval t, in MW. We assume that hydro units have the possibility of providing their full power output within a few minutes with essentially no cost. Hence, we neglect commitment decisions for hydro units.

The limits for the net output of each generation unit depend on its commitment status:

$$p_i^{\min} u_{it} \leq p_{it} \leq p_i^{\max} u_{it} \tag{19}$$

where $p_i^{\min}$ is the minimum net power output of thermal unit i when it is committed.

Hydro units also have limits for their net output, when they operate as generators, and for their net consumption, when they operate as pumps:

$$0 \leq p_{jt} \leq p_j^{\max} \tag{20}$$

$$0 \leq w_{jt} \leq w_j^{\max} \tag{21}$$

where $w_j^{\max}$ is the maximum net power consumption of hydro unit j when operating as a pump.

A relationship exists between start-up decisions and the commitment status of thermal units in consecutive time intervals:

$$s_{it} \geq u_{it} - u_{it-1} \tag{22}$$

We approximate the fuel consumed by each thermal unit in each time interval by an affine function of its net power output:

$$c_{it} = \alpha_i u_{it} + \beta_i p_{it} \tag{23}$$

The level of the upper reservoir of each hydro plant in each time interval depends on its operation:

$$l_{jt} = l_{jt-1} + I_{jt} - p_{jt} + \eta_j w_{jt} \tag{24}$$

where l_{jt} is the level of the upper reservoir of hydro plant j at the end of time interval t, in MWh, I_{jt} are the natural inflows received by the upper reservoir of j during t, in MWh, and η_j is the cycle (or pumping) efficiency of plant j, in p.u. An efficiency coefficient is necessary because not all the energy that is consumed to pump a certain amount of water is obtained when that amount water is released for production

As mentioned, this is a simplified representation and it does not consider features such as minimum up (down) time requirements for thermal units that start up (stop), or ramp limits for the power output of thermal units between consecutive time intervals.

In many power generation systems there are groups of identical thermal or hydro units. Although each of the units of one of these groups can be operated in an independent manner, it is reasonable to represent them in an aggregate manner, since it makes no difference which ones of identical units are in operation. This removes symmetries from the problem. Under such an aggregate representation, the commitment state of a group of N thermal units is represented with an integer variable ranging from 0 to N, rather than with N binary variables. We call each of these groups of units a generalized unit. In the VEAG case considered in this paper, the collection of 35 thermal units and 22 pumped-storage units may be aggregated into 14 generalized

thermal units and 8 generalized hydro units. Problem sizes, for an individual unit representation and an aggregated representation, are shown in Table 2.

	Rows	Columns	NonZero	Integer
Disaggregated units	29902	40152	104440	5712
Aggregated units	12769	16466	44163	2520

Table 2. Problem sizes for the VEAG UC problem.

4.2 Computational results

This section presents the computational results obtained when applying the proposed method to solve different stochastic instances of the numerical example. In addition to comparing the decomposition method with the direct solution of the problem we have explored the convenience of formulating large or small subproblems.

We have introduced uncertainty in the load profile by means of scenario trees, built from the original data with clustering techniques. In particular we have used scenario trees with 4, 7 and 12 scenarios (see Figure 4). An example of generation schedule is shown in Figure 5 for a four-scenario tree, which reflects the effect of pumped-storage units. In this example, with a thin style it has been depicted stochastic demand profile and with a large style it has been depicted the thermal production obtained. The load profile above the thermal production profile is covered with hydro generation. A thermal profile greater that the demand profile in off-peak hours indicates pumping in pumped-storage hydro units.

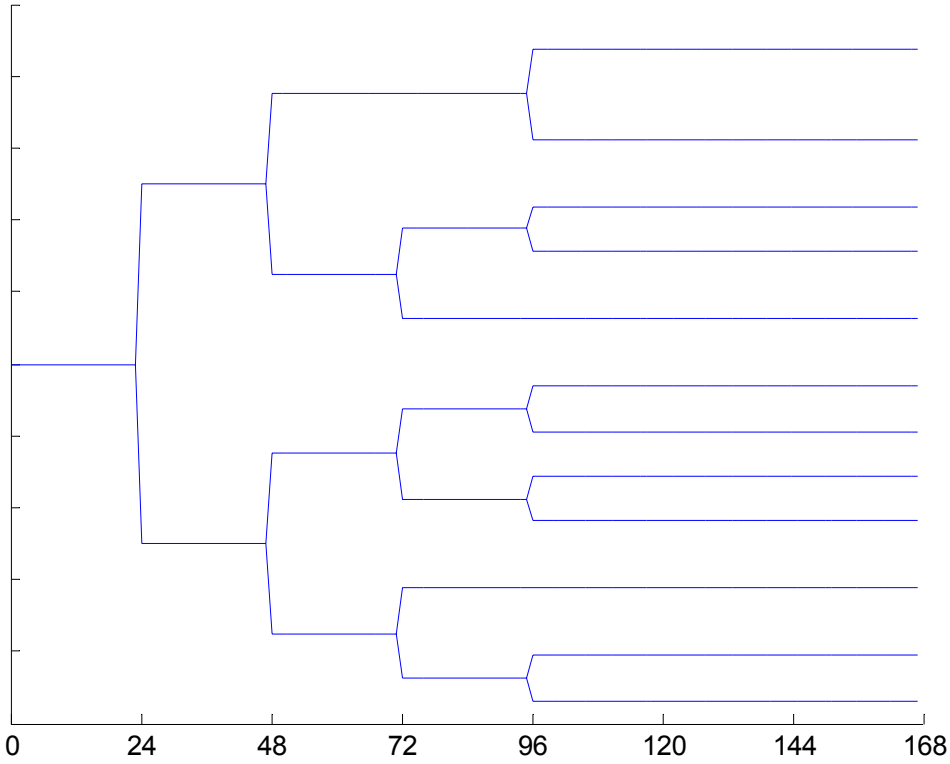

Figure 4. A scenario tree with twelve scenarios.

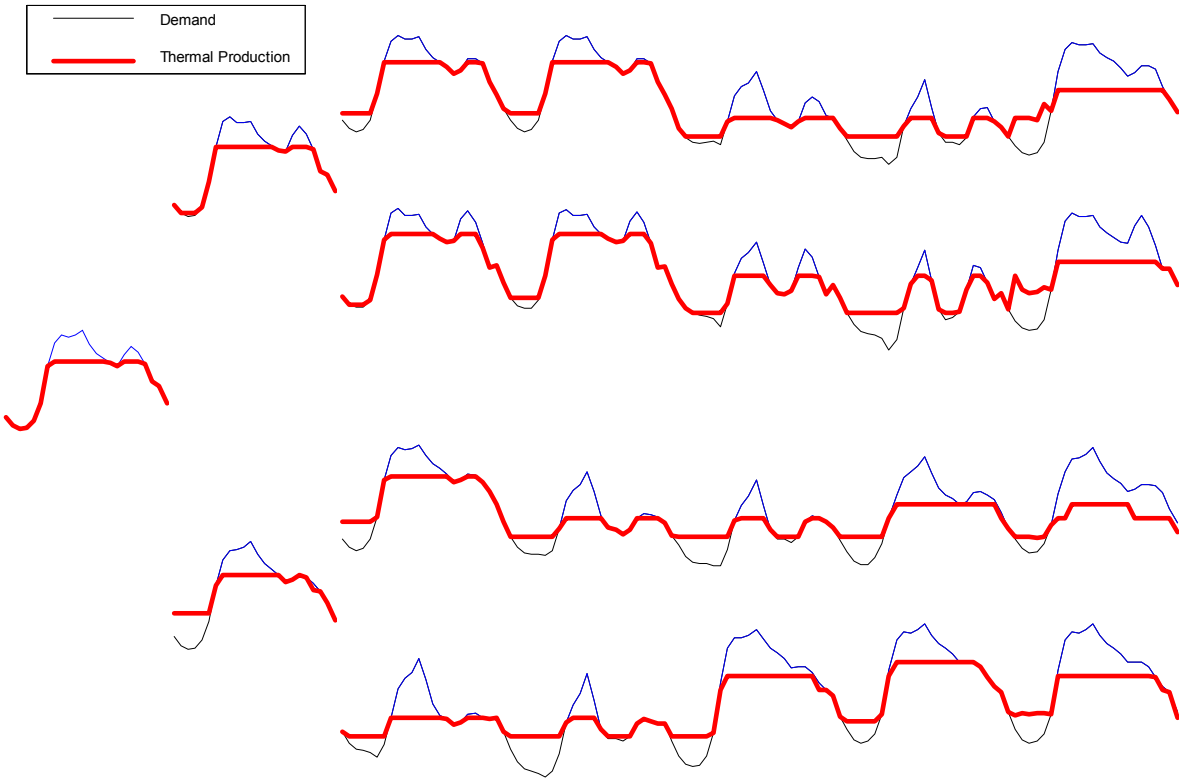

Figure 5. 4-scenario problem production schedule.

We have solved these numerical examples using the decomposition algorithm until a relative tolerance of 0.1 % was achieved. The sequential cut refinement method reached such tolerance in phase 2, so that phases 3 to 5 of the algorithm were not necessary. We have also carried out a direct solution of the problem to analyze the required time to achieve a similar tolerance. Table 3 presents the results for the VEAG system with disaggregated units (VD) and Table 4 the equivalent results for aggregated units (VA). In both cases we indicate the required time to achieve a certain tolerance.

	Direct		Our method	
VD 1 Scenario	Time (s)	4003	Time (s)	730
	Accuracy (%)	0.03	Accuracy (%)	0.07
VD 4 Scenarios	Time (s)	60000	Time (s)	3945
	Accuracy (%)	0.1	Accuracy (%)	0.06
VD 7 Scenarios	Time (s)	-	Time (s)	13730
	Accuracy (%)	-	Accuracy (%)	0.06

Table 3. Execution times for the VEAG system with disaggregated units.

	Direct		Our method	
VA 1 Scenario	Time (s)	51	Time (s)	30
	Accuracy (%)	0.02	Accuracy (%)	0.06
VA 4 Scenario	Time (s)	680	Time (s)	104
	Accuracy (%)	0.05	Accuracy (%)	0.04
VA 7 Scenarios	Time (s)	817	Time (s)	228
	Accuracy (%)	0.04	Accuracy (%)	0.05
VA 12 Scenarios	Time (s)	1840	Time (s)	329
	Accuracy (%)	0.04	Accuracy (%)	0.04

Table 4. Execution times for the VEAG system with aggregated units.

These results show that the solution of the problem with our extension of Benders algorithm together with the cut refinement method clearly outperforms the direct solution of the problem. Figure 6 confirms this by comparing computation times for the direct solution and the Benders' solution of the VA problem with different tree sizes.

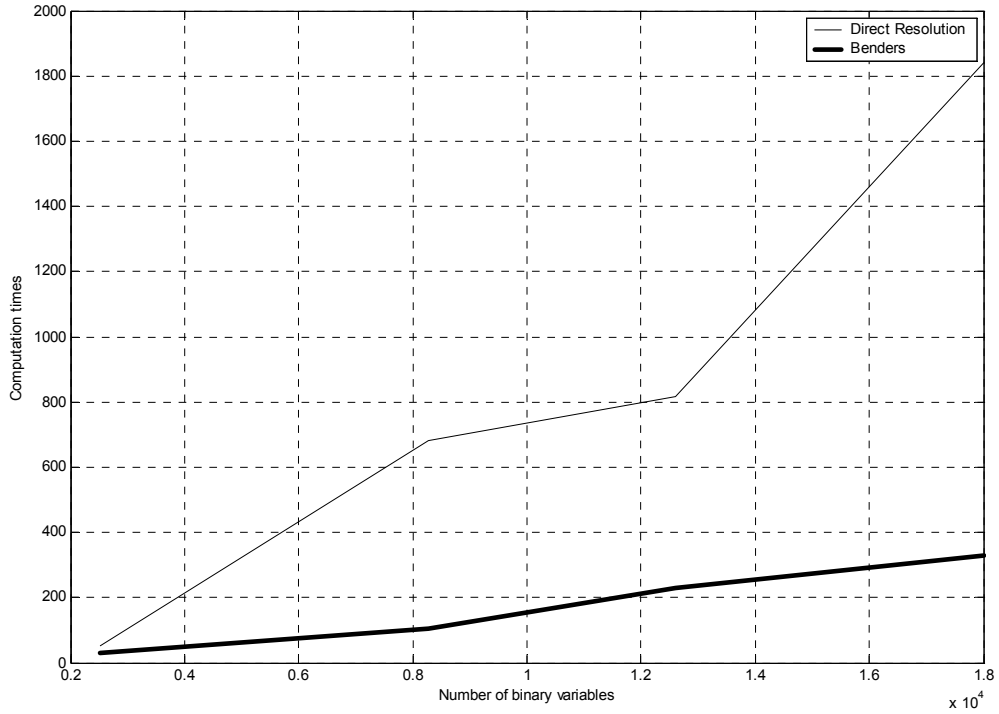

Figure 6. Execution times of the Benders method and the direct solution.

5 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 sequentially improving the approximation of the recourse function by computing computationally cheaper cuts prior to more expensive cuts.

Although this may seem a trivial result, the particular sequence of phases that we have proposed is an original contribution. We have illustrated the importance of the proposed sequential approach by means of an academic example that requires four of the five phases to reach the exact solution. We have also presented an application of the method to a real-size weekly stochastic UC problem. In this case, we have only had to execute two phases to reach a tolerance lower than 0.1 %. Our method clearly

outperforms the direct solution of the problem. An advantage of our method with respect to other solution approaches is that the solution obtained satisfies all the constraints of the original problem.

6 REFERENCES

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

BIRGE, J. R., F. LOUVEAUX. 1988. A Multicut Algorithm fo Two-Stage Stochastic Linear Programs. *European Journal of Operational Research* **34**.

BIRGE, J. R., F. LOUVEAUX. 1997. *Introduction to Stochastic Programming*. Springer Series in Operations Research. Springer, New York.

CARØE, C. C., R. SCHULTZ. 1998. Dual decomposition in stochastic integer programming. *Operations Research Letters* **24**: 37-45.

CARØE, C. C., J. TIND. 1998. L-Shaped Decomposition of Two-Stage Stochastic Programs with Integer Recourse. *Mathematical Programming* **83**.

CARPENTIER, P., G. GOHEN, J.-C. CULIOLI, A. RENAUD. 1996. Stochastic optimization of unit commitment: a new decomposition framework. *IEEE Transactions on Power Systems* **11(2)**: 1067-1073.

DENTCHEVA, D., W. RÖMISCH. 1998. Optimal power generation under uncertainty via stochastic programming, in *Stochastic Programming Methods and Technical Applications*, K. Marti, P. Kall, eds. Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin. **458:** 22-56.

DUPACOVÁ, J., G. CONSIGLI, S. W. WALLACE. 2000. Scenarios for Multistage Stochastic Programs. *Annals of Operations Research* **100**: 25-53.

FLIPPO, O. E., A. H. G. R. KAN. 1993. Decomposition in general mathematical programming. *Mathematical Programming* **60**: 361-382.

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

GEOFFRION, A. M. 1974. Lagrangean relaxation for integer programming. *Mathematical Programming Study* **2**: 82-114.

HOBBS, B. F., M. H. ROTHKOPF, R. P. O'NEILL, H.-P. CHAO. 2001. *The Next Generation of Electric Power Unit Commitment Models*. Kluwer Academic Publishers, Boston.

HOLMBERG, K. 1994. Cross decomposition applied to integer programmin problems: Duality gaps and convexifications in parts. *Operations Research* **42(4)**.

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

LAPORTE, G., F. LOUVEAUX. 1993. The Integer L-Shaped Method for Stochastic Integer Programs with Complete Reourse. *Operations Research Letters* **13**.

LEMARECHAL, C., A. RENAUD. 2001. A geometric study of duality gaps, with applications. *Mathematical Programming* **90**: 399-427.

MORTON, D. 1993. Algorithmic Advances in Stochastic Programming, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

NOWAK, M. P., W. RÖMISCH. 2000. Stochastic Lagrangian Relaxation applied to Power Scheduling in a Hydro-Thermal System under Uncertainty. *Annals of Operations Research*.

ROCKAFELLAR, R. T., R. J.-B. WETS. 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.

SHEBLÉ, G. B. 1994. Unit Commitment Literature Synopsis. *IEEE Transactions on Power Systems* **9(1)**: 128-135.

TAKRITI, S., J. R. BIRGE. 2000. Using Integer Programming to Refine Lagrangian-Based Unit commitment Solutions. *IEEE Transactions on Power Systems* **15(1)**.

TAKRITI, S., B. KRASENBRINK, L. S.-Y.-. WU. 2000. Incorporating Fuel constraints and Electricity Spot Prices into the Unit Commitment Problem. *Operations Research* **48(2)**: 268-280.

VALENZUELA, J., M. MAZUMDAR. 2003. Unit Commitment based on Hourly Spot Prices in the Electric Power Industry. *Operations Research* **51(6)**: 880-893.

VAN DER VLERK, M. 1995. Stochastic programming with integer recourse. University of Groningen**.**

VAN SLYKE, R. M., R. WETS. 1969. L-Shaped Linear Progams with Applications to Optimal Control and Stochastic Programming. *SIAM Journal on Applied Mathematics* **17(4)**: 638-663.

WETS, R. J.-B. 1996. Challenges in stochastic programming. *Mathematical Programming* **75**: 115-135.