Algorithmic Operations Research Vol.4 (2009) 76–85

A Solution Approach for Two-Stage Stochastic Nonlinear Mixed Integer Programs

Patrizia Beraldi, Maria Elena Bruni, and Domenico Conforti

Department of Electronics, Informatics and Systems, Via P. Bucci Cubo 41/C, University of Calabria 87036 Rende (Cs) Italy

Abstract

This paper addresses the class of nonlinear mixed integer stochastic programming problems. In particular, we consider two-stage problems with nonlinearities both in the objective function and constraints, pure integer first stage and mixed integer second stage variables. We exploit the specific problem structure to develop a global optimization algorithm. The basic idea is to decompose the original problem into smaller manageable optimization subproblems and coordinate their solutions by means of a Branch and Bound approach. Preliminary computational experiments have been carried out on a stochastic version of the Trim Loss problem.

Key words: Stochastic Programming, Mixed Integer Nonlinear Programming, Trim Loss Problem, Lagrangian Decomposition, Branch and Bound.

1. Introduction

Stochastic integer programming (SIP) represents one of the most challenging area in the field of the modern stochastic optimization. When nonlinearities are present in the objective function and/or constraints we have to deal with stochastic nonlinear integer problems. Such class of problems provides a very powerful modeling tool for decision making integrating the expressive power of nonlinear integer deterministic models with the key issue of uncertainty affecting all the real-life applications.

The design of batch plants [1], the synthesis of process [2], the design of distillation sequences [3], the minimization of waste in paper cutting [4], the optimization of core reload patterns for nuclear reactors [5], are just few examples of decision problems involving integer variables and nonlinear functions. The interested readers are referred to [6] for a comprehensive survey of nonlinear integer applications. It is worthwhile noting that, in almost all the applications mentioned above some of the parameters are not known in advance, making of little value the recommendations provided by the solution of the deterministic problems. More accurate models should take explicitly into account uncertainty. Here we mention two applications, the process design under uncertainty [7], and the airline crew scheduling problem [8], modelled by means of the stochastic programming framework.

Although the past decade has witnessed great effort in achieving theoretical and methodological development of the stochastic (linear) integer problems (see [9] and the references therein), the nonlinear integer case has received very limited attention. This partly derives from its challenging feature due to the combinatorial nature and the nonlinearity inherent in the problem.

In [10] Wei and Realff have proposed two algorithms (the optimality gap and the confidence level method) to solve stochastic mixed-integer nonlinear convex programming problems. Their methods are wrapped around a traditional approach for deterministic problems, the outer approximation method. The stochastic version of this approach solves, at each iteration, a stochastic nonlinear subproblem with fixed integer variables to provide an upper bound and a mixed-integer nonlinear problem to provide a lower bound and new values for the integer variables. A branching algorithm has been proposed by Birge and Yen in [8]. The method was designed for a specific application in the field of the air crew scheduling. Exploiting the specific problem structure, the algorithm branches simultaneously on multiple variables without invalidating the optimality conditions. Norkin et al. [11] developed a branch and bound algorithm that makes use of stochastic upper and lower bounds with almost sure convergence. An algorithm for the parametric solution of mixed integer nonlinear models arising in the context of process synthesis problems under uncertainty has been proposed in [12]. The method, based on the outer approximation/equality relaxation algorithm, involves the iterative solution of nonlinear subproblems and a parametric integer programming master problem. Different integration schemes for the approximation of the expectancy have been proposed in [13–16]. More recently, an integration of the sampling based L-shaped method with a reweighting concept has been used to solve stochastic nonlinear problems. The central idea is to reduce the computations at the sub-problem solution stage by using a reweighting scheme to bypass the nonlinear model computations.

All the contributions mentioned above deal with the case of continuous random variables. In this paper we propose a method for the global optimization of stochastic nonlinear integer problems with discrete distributions. The basic idea underlying the approach is to exploit the problem structure to decompose it into smaller manageable optimization subproblems and coordinate their solutions by means of a branch and bound approach. The proposed method belongs to the class of dual decomposition methods applied by Carøe and Schultz in [18] for the case of two-stage stochastic linear (mixed) integer problems. Our contribution generalizes the results on decomposition methods to the stochastic nonlinear integer case and, more importantly, introduces the incremental subgradient approach as a key factor in assessing the efficiency of the solution of the Lagrangian dual problems.

The remainder of the paper is organized as follows. Section 2 introduces the two-stage stochastic nonlinear (mixed) integer problem. In Section 3 the proposed solution method is presented by devoting particular attention to the solution of the Lagrangian dual problem. Section 4 presents and discusses some preliminary computational experiments carried out on a stochastic version of the Trim Loss problem. Conclusions and further research directions are illustrated in the last section.

2. Problem formulation

Let us consider a given probability space $(\Omega, \Im, \mathbb{P})$. For each element ω of the sample space Ω, we denote by $\xi(\omega)$ a finite dimensional random vector and by $\mathbb{E}_\xi$ the mathematical expectation with respect to ξ. Later on, we shall focus our attention on the following two-stage nonlinear (mixed) integer model:

$$
\begin{aligned}
\min \;& f^1(x) + \mathbb{E}_\xi Q(x, \xi(\omega)) \\
& g_i^1(x) = 0 \quad i = 1, \ldots, \bar{m}_1 \\
& g_i^1(x) \leq 0 \quad i = \bar{m}_1 + 1, \ldots, m_1 \\
& x \in \mathbb{Z}_+^{n_1}
\end{aligned}
\tag{1}
$$

where for a given realization $\omega \in \Omega$, $Q(x, \xi(\omega))$ is the optimal value of the second-stage (recourse) problem:

$$
\begin{aligned}
Q(x, \xi(\omega)) = \min \;& f^2(y(\omega), \omega) \\
& h_i^2(x, \omega) + g_i^2(x, y(\omega), \omega) = 0, \; i = 1, \ldots, \bar{m}_2 \\
& h_i^2(x, \omega) + g_i^2(x, y(\omega), \omega) \leq 0, \; i = \bar{m}_2 + 1, \ldots, m_2 \\
& y \in \mathbb{R}^{n_2 - t_2} \times \mathbb{Z}^{t_2}
\end{aligned}
\tag{2}
$$

where all functions $f^1, f^2, g^1, g^2, h^1, h^2$ are general nonlinear functions and $f^2(\cdot, \omega), g^2(\cdot, \omega), h^2(\cdot, \omega)$ are measurable in ω for any fixed first argument. According to the stochastic programming nomenclature, variables x denote the first stage decisions which need to be determined prior to the realization of the uncertain parameters ω, whereas variables y represent the recourse decisions that can be taken after uncertainty is disclosed.

There is a severe shortage of nice properties such as convexity and continuity in two-stage nonlinear integer problems. This is mainly due to the integer restrictions. If only the first stage variables are integer, the properties of the recourse function are the same as in the continuous case. In the continuous nonlinear case if f, h are convex and g is affine for all ξ, the problem is convex. When integrality restrictions are present in the second stage, even for the linear case the recourse function is in general nonconvex. The optimization of such a complex objective function poses severe difficulties.

In the following, we shall assume that the uncertain parameter ω follows a discrete distribution with finite support $\Omega = \{\omega_1, \omega_2, \cdots, \omega_S\}$. Each realization (scenario) $s = 1, \ldots, S$ has an associated probability p_s. We observe that discrete distributions arise frequently in applications, either directly, or as empirical approximations of the underlying probability distribution. Furthermore, as shown in [19] if the random parameters have a continuous distribution the optimal solution of the problem can be approximated within any given accuracy by the use of discrete distributions. Under the assumption of discrete probability space, problem (1)-(2) can be restated as follows:

$$
\begin{aligned}
\min & f^1(x) + \sum_{s=1}^{S} p_s f^2(x, y_s, \xi_s) \\
& g_i^1(x) = 0 \quad i = 1, \ldots, \bar{m}_1 \\
& g_i^1(x) \leq 0 \quad i = \bar{m}_1 + 1, \ldots, m_1 \\
& h_i^2(x, \xi_s) + g_i^2(x, y_s, \xi_s) = 0, \; i = 1, \ldots, \bar{m}_2
\end{aligned}
$$

$$s = 1, \dots, S \quad (3)$$
$$h_i^2(x, \xi_s) + g_i^2(x, y_s, \xi_s) \le 0,\ i = \bar{m_2} + 1, \dots, m_2 \quad s = 1, \dots, S$$
$$x \in \mathbb{Z}^{n_1}, y_s \in \mathbb{R}^{n_2 - t_2} \times \mathbb{Z}^{t_2} \quad s = 1, \dots, S$$

Problem (3) is a large-scale structured nonlinear (mixed) integer model with $(n_1 + n_2 \times S)$ variables and $(m_1 + S \times m_2)$ nonlinear constraints. Thus, at least in principle, standard programming techniques for deterministic nonlinear (mixed) integer problems implemented in general-purpose software, could be applicable. Despite the attractiveness of the claim, standard software does not perform well for the case of linear functions, and, a fortiori we expect worse performance for nonlinear problems. In order to face this computational challenge, the key issue is to exploit the specific problem structure to design an efficient solution method. In the next section we present our proposal.

3. The solution approach

The proposed solution method belongs to the class of dual decomposition methods proposed in [18] for the case of stochastic linear integer problems. For a very good review on decomposition methods for stochastic programming the interested reader is referred to [20]. The basic idea is to exploit the problem structure to decompose the original problem into smaller manageable optimization subproblems and coordinate their solutions by means of a branch and bound scheme.

Our approach differs from earlier works based on similar ideas. First of all it generalizes the results on decomposition methods to the stochastic integer nonlinear programs and, secondly, introduces the incremental subgradient approach as a key factor in assessing the efficiency of the solution of the Lagrangian dual problem.

Let us consider problem (3). It is easy to recognize that the model presents a block diagonal structure, where optimization problems pertaining to separate scenarios are tied together by means of the global variables x that can be viewed as complicating variables. By applying a variable splitting scheme [21], which introduces copies $x_1, \dots, x_S$ of the first-stage variable x and adds simple linking constraints, problem (3) can be reformulated as:

$$\min \sum_{s=1}^{S} p_s[f^1(x_s) + f^2(x_s, y_s, \xi_s)] \quad (4)$$
$$g_i^1(x_s) = 0,\ i = 1, \dots, \bar{m_1},\ s = 1, \dots, S \quad (5)$$
$$g_i^1(x_s) \le 0,\ i = \bar{m_1}+1, \dots, m_1,\ s = 1, \dots, S \quad (6)$$
$$h_i^2(x_s, \xi_s) + g_i^2(x_s, y_s, \xi_s) = 0,\ \ i = 1, \dots, \bar{m_2} \quad s = 1, \dots, S \quad (7)$$
$$h_i^2(x_s, \xi_s) + g_i^2(x_s, y_s, \xi_s) \le 0, \quad i = \bar{m_2}+1, \dots, m_2, s = 1, \dots, S \quad (8)$$
$$x_s = x_{s+1} \quad s = 1, \dots, S-1 \quad (9)$$
$$x_s \in \mathbb{Z}_+^{n_1}, y_s \in \mathbb{R}^{n_2 - t_2} \times \mathbb{Z}^{t_2} \quad s = 1, \dots S \quad (10)$$

Constraints (9) are aimed at guaranteeing the nonanticipativity principle, which states that the first-stage decisions are scenario-invariant since they do not depend on the scenario which will prevail in the second stage. This constraint can also be represented as $\sum_{s=1}^{S} A_s x_s = 0$, where A_s are matrices of suitable dimensions. The Lagrangian relaxation with the respect to the nonanticipativity constraints leads to the following problem:

$$D(\lambda) = \min \sum_{s=1}^{S} p_s[(f^1(x_s) + f^2(x_s, y_s, \xi_s)] + \sum_{s=1}^{S} \lambda(A_s x_s) \quad (11)$$
$$g_i^1(x_s) = 0,\ i = 1, \dots, \bar{m_1},\ s = 1, \dots, S \quad (12)$$
$$g_i^1(x_s) \le 0,\ i = \bar{m_1}+1, \dots, m_1,\ s = 1, \dots, S \quad (13)$$
$$h_i^2(x_s, \xi_s) + g_i^2(x_s, y_s, \xi_s) = 0 \ \ i = 1, \dots, \bar{m_2}, \quad s = 1, \dots, S \quad (14)$$
$$h_i^2(x_s, \xi_s) + g_i^2(x, y_s, \xi_s) \le 0 \ i = \bar{m_2}+1, \dots, m_2, \quad s = 1, \dots, S \quad (15)$$
$$x_s \in \mathbb{Z}_+^{n_1}, y_s \in \mathbb{R}^{n_2 - t_2} \times \mathbb{Z}^{t_2} \ \ s = 1, \dots S \quad (16)$$

where λ is an appropriately dimensioned vector of Lagrangian multipliers. The variable splitting method was originally applied in conjunction with Lagrangian relaxation [22] to optimization problems with ‘hard’ and ‘soft’ set of constraints and it is equivalent to what is termed Lagrangian Decomposition in [23]. Carøe and Schultz [18] and Hemmecke and Schultz [24] used a similar approach to obtain bounds for two-stage linear integer problems. We also mention Takriti and Birge [25]. For an impression on Lagrangian approaches for multistage stochastic integer programming developments we refer to Römisch and Schultz [26].

The main advantage of the reformulation (11)-(16) comes from the perfect decomposability of the original problem into S independent subproblems:

$$D(\lambda) = \sum_{s=1}^{S} D_s(\lambda) \tag{17}$$

where

$$D_s(\lambda) = \min\{p_s[(f^1(x_s) + f^2(x_s, y_s, \xi_s)] + \lambda(A_s x_s) : (x^s, y^s) \in X^s\}$$

and X^s denotes the set of constraints for scenario s. It is well known that the Lagrangian dual

$$\max_{\lambda} D(\lambda) \tag{18}$$

provides a lower bound on the optimal value of problem (4)-(10). In addition, if for some choice of λ the scenario solutions of the Lagrangian relaxation (x_s, y_s) coincide in their first-stage components, then they are also optimal. In order to enforce the relaxed nonanticipativity constraints, as in [18], we have used the Lagrangian dual as bounding rule within a branch and bound procedure.

Let us denote by L the list of candidate problems l together with an associated lower bound z_{LD}. The outline of the algorithm is as follows:

Step1 (Inizialization). Set $\bar{z} = +\infty$ and let L contain problem (4)-(10).

Step2 (Termination). If L=∅ then the solution (x, y) that yielded $\bar{z}$ is optimal.

Step3 (Node Selection). Select and delete a problem l from L, solve the corresponding Lagrangian dual and let $z_{LD}(l)$ denote its corresponding optimal value. If l is infeasible go to Step 2.

Step4 (Bounding). If $z_{LD}(l) \geq \bar{z}$ go to Step 2. Otherwise, if the scenario solutions x_s are identical, update $\bar{z}$ and delete from L all subproblems with $z_{LD}(l) \geq \bar{z}$. Go to Step 2. Else if the scenario solutions differ, determine a candidate feasible first-stage solution $\bar{x}^R$, update $\bar{z}$ and delete from L all problems with $z_{LD}(l) \geq \bar{z}$. Go to Step 5.

Step5 (Branching). Select a component x_i of x and add to L two new problems obtained from l by adding the constraints $\bar{x}_i \leq \lfloor \bar{x}_i \rfloor$ and $\bar{x}_i \geq \lfloor \bar{x}_i + 1 \rfloor$, respectively. Go to Step 2.

At Step 4, the candidate for feasible first-stage solution x can be determined by using various heuristic ideas. A possibility is to combine the average

$$\bar{x} = \sum_{s=1}^{S} p_s x_s \tag{19}$$

with some rounding heuristic to fulfill the integrality restrictions. If the original problem is feasible, since the number of nodes generated and explored in the branch and bound tree is finite, the algorithm terminates after finitely many steps. The optimality of the solution follows from the validity of the lower and upper bounds used.

3.1. *The solution of the Lagrangian dual problem*

The efficient solution of the Lagrangian dual problems within the branch and bound scheme represents a critical issue because of the problem nature (nondifferantiable concave) and the number of times the solution process has to be performed. To this aim we have designed an incremental subgradient method which exploits the specific structure of our problem. The incremental subgradient method was originally proposed in [27] for minimizing a convex function expressed as sum of a large number of component functions. Such a method is similar to the standard subgradient method [28]. The main difference is that the multiplier vector update is performed after each subgradient component computation. Thus, the multiplier vector is changed incrementally with intermediate adjustment of the variables after processing each component function. The basic steps of the method are as follows. The subgradient of (11) at λ is $g(\lambda) = \sum_{s=1}^{S} A_s x_s(\lambda)$, where $x_s(\lambda)$ are optimal solutions of the scenario subproblems. The subgradient is a vector of dimension $n_1(S-1)$ and is the sum of $g_i(\lambda)$, where $g_i(\lambda)$ is a subgradient of D_i at λ.

Let us denote by the superscript k the iteration counter of the standard subgradient method. Each step is a subgradient iteration for a single component function (single scenario in our setting), and there is one step per component function. Thus, an iteration can be viewed as a cycle of S subiterations.

At a generic iteration k of the subgradient method $\lambda^{k+1} = \phi_S^k$, where ϕ_S^k is obtained after the $|S|$ steps

$$\phi_s^k = [\phi_{s-1}^k - \alpha_k g_{s(\lambda)}^k], \quad s = 1, \ldots, S \tag{20}$$

and

$$\phi_0^k = \lambda^k \tag{21}$$

The updates described in (20) are referred to as the S subiteration of the kth cycle. In all subiterations of a cycle we use the same stepsize α_k. Such incremental approach allows us to carry information from one subiteration to the next, thus avoiding the need to work out the multiplier vector from scratch.

We observe that the rows of the matrix $A = [A_1, A_2, \ldots, A_S]$ have only two nonzero components equal to 1 and -1. Giving the particular structure of the nonanticipativity constraints, at each subiteration of the method sketched above, the subgradient vector $g_{i(\lambda)}$ is worked up by taking into account the term relative to one scenario $A_s x_s(\lambda)$. For the first scenario the only nonzero in the subgradient vector are those relative to the first n_1 rows of the matrix A_1 which has the form:

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & 1 \\ 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

and thus has a diagonal block of 1 and all other elements zero. This implies that only the portion of multiplier λ associated with the first scenario will be changed during the first update. For the second scenario the matrix has two diagonal blocks due to the fact that variables associated with the second scenario are present in two constraints of type (9). A_2 has the form:

$$\begin{pmatrix} -1 & 0 & \dots & 0 \\ 0 & \ddots & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & -1 \\ 1 & 0 & \dots & 0 \\ 0 & \ddots & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}.$$

In this case only the components of the multiplier λ associated with the first and the second scenario will be changed. Similar considerations can be drawn for the remaining scenarios. For all the scenarios except that for the first one, only two components of the multiplier vector are updated with the rule (20) because only two blocks have nonzero elements. Thus, once the subproblem for a scenario s is solved and the multiplier vector has been updated according with (20),we keep this value fixed for the remaining $S-1$ subiteration. This case allows us to deal with a restricted dimension of the dual vector λ in the subproblem, speeding the solution process.

It can be verified that the order used for processing the component functions $D_s(\lambda)$ can significantly affect the rate of convergence of the method. A randomized version of the incremental subgradient method has been proposed in [28], where the component function to be processed is chosen randomly. In our case since each component function $D_s(\lambda)$ has an associated probability, namely the probability of scenario s, we select the scenario to be processed according to the probability distribution.

Clearly, an efficient implementation of the proposed algorithm needs to address a number of issues related to the presence of a large number of alternative scenarios. It is not difficult to recognize that the algorithm can potentially become very intensive from the computational point of view. Although the best approach is likely to be problem dependent, some general recommendations are noteworthy. Firstly, the solution to optimality of the Lagrangian dual comes at a computational cost, so it can result in a decrease in the number of iterations to convergence, but in an increase in the solution time. Instead of solving the Lagrangian dual to optimality, it is beneficial to stop the solution process as soon as the Lagrangian value rises above the best known upper bound $\bar{z}$. Similar ideas have been used in [29] and [30]. Secondly, by exploiting the solutions of the incremental subgradient subproblems, an effective early branching strategy has been designed to reduce the computational burden at each node of the branch and bound tree. The early branching scheme exploits the quality of the solution in term of its nonanticipativity gap. We define $\tau = \sum_{s=1,\dots,S-1} |x_{(s+1)} - x_{(s)}|$ as a measure of the nonanticipativity of the solution. To increase the impact of variable branching on the enforcement of the nonanticipativity gap, a variable $\hat{j} =$ argmax τ_j is selected for early branching when $\tau_{\hat{j}} \geq \varepsilon$, where the acceptable nonanticipativity degree ε is an arbitrary parameter which depends on the problem at hand.

The efficiency of the proposed method heavily relies on the ability to solve nonlinear (mixed) integer subproblems (17). We observe that these subproblems have to be solved a number of times depending on the number of iteration step to solve (18). In order to improve efficiency, we have implemented a warm start procedure. Since subproblems generated at a given node of the branch and bound tree differ only in a bound constraint from the father, dual multipliers can be passed from the parent to the child nodes.

4. Numerical Illustration

In contrast to the linear (mixed) integer case, no test problems to use as benchmark have been proposed for the nonlinear counterpart yet. Thus, in order to test the efficiency of the proposed solution approach, we have faced the problem of generating meaningful instances. To this aim, we have considered a stochastic version of a well-known deterministic problem: the Trim Loss problem (see, for example, [31] for a general description of the problem). The stochastic Trim Loss problem represents a very interesting generalization of its deterministic counterpart since it explicitly incorporates uncertainty in one of the most studied problems in the field of the manufacturing applications. Because of its relevance, many methods both exact ([32–34]) and heuristic ([35,36]) have been proposed for its solution over the last decades.

Consider the problem of cutting different products $i = 1, \ldots, I$ from raw materials (e.g. paper rolls). Each type of product i can be cut by means of different cutting patterns $j = 1, \ldots, J$, each defined by the position of the knives. For each product i, a certain width b_i and a demand d_i are defined. The change of a cutting pattern involves a cost C_j since the cutting machine has to be stopped before repositioning the knives. In addition, typically it is not possible to cut out an order, specified by the demands, without throwing away some of the raw material. Roughly speaking, the problem consists of determining the cutting scheme which allows to satisfy the customers demands minimizing the total cost. The cutting pattern is defined by specifying the use of a given pattern by means of the binary variable y_j, the number of products i in pattern j by n_{ij} and the number of repeats m_j of a pattern j .

The stochastic version of the problem has been defined by explicitly incorporating into the deterministic model the main source of uncertainty which is related to the demands. We have assumed that the uncertain demands are represented by discrete random variables with a finite number of realizations d_i^s each occurring with probability p_s, $s = 1, \ldots, S$. The two-stage structure has been suggested by the nature of the problem: decisions concerning the existence of a pattern need to be taken in advance in order to set the cutting knives, whereas the number of repeats can be decided when additional information is available. The formulation of the two-stage Trim Loss problem is as follows:

$$\min \sum_{s=1}^{S} \sum_{j=1}^{J} (C_j y_j + p_s c_j m_{js}) \tag{22}$$

$$(B_{\max} - \Delta) y_j \le \sum_{i=1}^{I} b_i n_{ij} \le B_{\max} y_j \ \forall j \tag{23}$$

$$y_j \le \sum_{i=1}^{I} n_{ij} \le N_{max} y_j \ \forall j \tag{24}$$

$$\sum_{j=1}^{J} m_{js} n_{ij} \ge d_i^s \quad \forall\, i,\ \forall s \tag{25}$$

$$y_j \le m_{js} \le M y_j \ \forall j,\ \forall s \tag{26}$$

$$\sum_{j=1}^{J} m_{js} \ge \max\left(\left\lceil \frac{\sum_{i=1}^{I} d_i^s}{N_{max}} \right\rceil, \left\lceil \frac{\sum_{i=1}^{I} d_i^s b_i}{B_{\max}} \right\rceil \right) \ \forall s \tag{27}$$

$$y_{(j+1)} \le y_j \quad j = 1, \ldots, J-1 \tag{28}$$

$$m_{(j+1)s} \le m_{js} \quad j = 1, \ldots, J-1,\ \forall s \tag{29}$$

$$y_j \in \{0,1\} \ \forall j$$

$$m_{js} \in \mathbb{Z},\ \forall j, \quad \forall s \tag{30}$$

$$n_{ij} \in \mathbb{Z},\ \forall i, \quad \forall j. \tag{31}$$

where y_j andn_{ij} are first-stage variables, whereas m_{js} denote the second stage variables. In the above formulation, for each cutting pattern j, M represents an upper bound on number of repeats, whereas $B_{\max}$, Δ and N_{max} denote the maximum width allowed and the width tolerance for cutting patterns, and a physical restriction of the number of knives that can be used in the cutting process, respectively. Constraints (23) prevent the patterns to exceed the given width limits, constraints (24) limit the maximum number of products that can be cut from a pattern, for each scenario s, constraints (25) impose the satisfaction of the customer demands. Constraints (26) relate the binary variables y_j to the cutting pattern. Constraints (27) impose a lower bound on total number of patterns made, whereas (28) and (29) are precedence constraints used to reduce degeneracy. It is worth noting that the stochastic formulation has been derived from the standard deterministic one used in [6] and [37]. To our knowledge the stochastic Trim Loss problem has not been previously addressed in the literature. The bilinear constraints determine the nonconvex integer nature of the problem. Since the number variables $[J + J \times I + J \times S]$ and constraints $[4 \times J + I \times S + 3(J \times S) + S]$ is related to the number of scenarios, depending on S, the problem becomes intractable.

Our preliminary computational experiments have been carried out on set of instances derived from the deterministic models whose parameters are reported in Table 1. By varying the number of scenarios, we have defined 12 instances, whose sizes, measured in terms of number $|V|$ of variables and $|C|$ of constraints are reported in Table 2.

Table 1

Problem parameters

	I	J	C_j	c_j	N_{max}	$B_{\max}$	M	Δ	b_i
TL2	2	2	1	$\frac{1}{10}j$	5	1900	3	200	$\{330, 360\}$
TL4	4	4	1	$\frac{1}{10}j$	5	1900	15	200	$\{360, 385, 415, 330\}$
TL5	5	5	1	$\frac{1}{10}j$	5	2200	15	200	$\{330, 360, 370, 415, 435\}$

Table 2

Test Problems dimensions

Problem	S	$\|V\|$	$\|C\|$
TL2_50	50	106	458(100)
TL2_100	100	206	908(200)
TL2_150	150	306	1358(300)
TL2_200	200	406	1808(400)
TL4_50	50	220	866(200)
TL4_100	100	420	1716(400)
TL4_150	150	620	2566(600)
TL4_200	200	820	3416(800)
TL5_50	50	280	1070(250)
TL5_100	100	530	2120(500)
TL5_150	150	780	3170(750)
TL5_200	250	1030	4220(1000)

Our prototypal algorithm has been implemented in C++ and uses Lindo Api as callable library [38] to solve the different subproblems within the branch and bound scheme. The choice of this software has been motivated by the consideration that it provides a provably global optimal solution and is one of the fastest and most robust available global solvers. On the contrary, the main disadvantage derives from the interface style which is based on the 'instruction list' input format. Such a notation can be very time consuming even for problems of medium size. However, the nature of the method which works on independent scenario subproblems facilitates this task. Although our implementation uses Lindo Api, any global optimization software with callable library and standard interface can serve this purpose. We observe that the competitive advantage of the proposed method over straightforward use of standard solvers applied to the deterministic equivalent problems derives from the exploitation of the stochastic problem structure.

The performance of the implemented algorithm has been evaluated by measuring the solution time (CPU time) and the number of outer iterations (NIter). We report in Figures 1-3 the CPU time in seconds for solving TL2, TL4 and TL5, respectively, for different number of scenarios. The analysis of results show that doubling the number of scenarios does not produce a substantial impact on the solution time. This behavior highlights the benefits of the criteria implemented in our algorithm. The selection of appropriate branching variables also plays a key role in ensuring high performances of the algorithm. In particular for the Trim Loss problem, the y variables are assigned the highest priority, the m variables come next, and the n variables follow. The number of major iterations performed by our algorithm is reported in Figure 4.

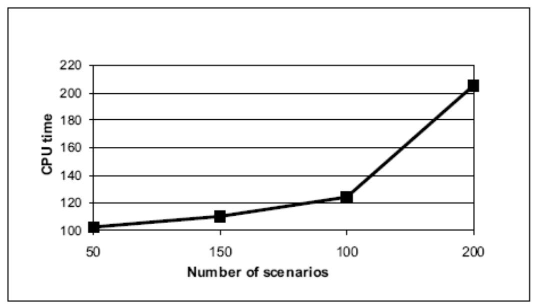

Fig. 1. CPU time in seconds for the TL2 as function of the number of scenarios.

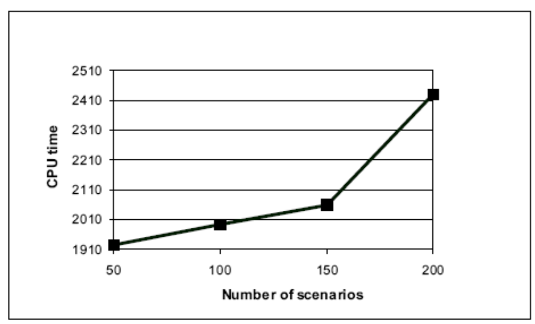

Fig. 2. CPU time in seconds for the TL4 as function of the number of scenarios.

Other experiments have been carried out to compare our algorithm with the standard solver (see Table 3). To this end, a time limit on the running time has been fixed at 3600 seconds for the instances TL2 and TL4

Table 3

General-purpose solver versus our algorithm for TL2_5 and TL2_10

Problem	General-purpose Solver		Decomposition algorithm	
	CPU time	NIter	CPU time	NIter
TL2_5	72	138941	2	3383
TL2_10	11	11085	3	3877

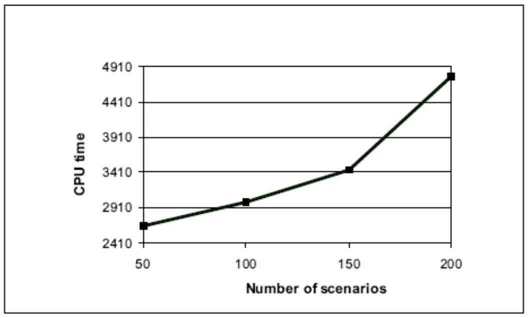

Fig. 3. CPU time in seconds for the TL5 as function of the number of scenarios.

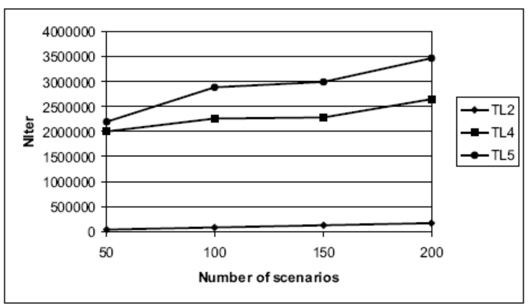

Fig. 4. Number of major iterations

and to 5000 seconds for the instance TL5. We observe, however, that such a comparison was not very significant: the standard solver was not able to solve none of the instances within the fixed limit, but the TL2 with a very limited number of scenarios. This behavior can be explained by observing that our method fully exploits the structure of the considered problem.

Finally, in Figure 5 we have compared the CPU time for the test problems TL2, TL4 and TL5 with the same number of scenarios. To sum up, on the basis of the numerical results we can observe that:

- For all the test problems the proposed algorithm offers significant advantages over the general-purpose solver. Thus, decomposition seems to be the best way to tackle this kind of problem;
- Notwithstanding the problem of finding a globally

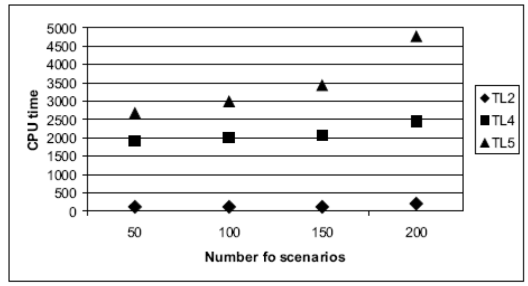

Fig. 5. CPU time in seconds as function of the

optimal solution of a nonconvex problem is a NP-hard task and the time to find a global optimum may increase exponentially with problem size, our algorithm is very efficient in practice;
- The instance TL2 with up to 1202 variables, 3100 linear constraints, and 400 nonlinear constraints can be solved in less than 5 minutes as shown in Figure 1;
- Increasing the number of first stage variables makes the problem more difficult for both the general-purpose solver and our algorithm;
- Increasing the number of patterns variables while maintaining fixed the number of scenarios correspond to an increase in the CPU time which is more evident for the instances TL4 and TL5. As expected, the computing time of the algorithm increases with the inherent difficulty of the deterministic instances;
- The algorithm is quite insensitive to the scenarios growth although the problem dimension depends nonlinearly on the number of scenario S. This shows the effectiveness of the implementation issues addressed in our prototypal algorithm.

5. Conclusion

In this paper we have proposed a solution approach for the class of two-stage stochastic nonlinear (mixed) integer problems. The huge size of the deterministic equivalent formulation makes the solution process overwhelming for general-purpose software. The proposed approach belongs to the class of dual decomposition methods. In particular, the relaxation of the nonanticipativity constraints makes the original problem separable into independent scenario subproblems. The coordination of the different scenario solutions is performed by means of a branch and bound approach where the solution of the Lagrangian dual is used as bounding rule. The solution of the corresponding nondifferentiable concave subproblems is performed by an incremental subgradient method which exploits the specific problem structure. The preliminary computational results carried out on a stochastic version of the Trim Loss problem are very encouraging suggesting that the proposed algorithm needs to be investigated further to identify additional properties and application areas. Furthermore, the proposed approach seems to be suitable to be implemented on a parallel computational environment. Infact, the solution of the scenario subproblems can be carried out in parallel by partitioning the workload among the available processors. The design of an efficient parallel implementation represents an ongoing research activity.

References

[1] Kocis G.R. and Grossmann,I.E.,1988, Global optimization of nonconvex mixed-integer nonlinear programming problems in process synthesis, *Ind. Eng. Chem. Res.*, **27 (8)**, 1407–1421.

[2] Duran, M.A. and Grossmann,I.E., 1986, An outer-approximation algorithm for a class of mixed-integer nonlinear programs, *Mathematical Programming*, **36**, 307–339.

[3] Aggarwal A., and Floudas,C.A., 1990, Synthesis of general distillation sequences: nonsharp separation, *Comput. Chem. Eng.*, **14 (6)**, 631–653.

[4] Harjunkoski, I., Westerlund, T., Porn, R., Skrifvars, H.,1998, Different transformations for solving nonconvex trim loss problems by MINLP, *European Journal of Operational Research* , **105**, 594–603.

[5] Quist, A.J., de Klerk, E., Roos, K., Terlaky, T., van Geemert, R., Hoogenboom, J.E., and Illes, T.,1999, Finding optimal nuclear reactor core reload patterns using nonlinear optimization and search heuristics, *Engineering Optimization*, **32**, 143-176.

[6] Floudas, C. A, Pardalos, P. M., Adjiman, C .S., Esposito, W. R., Gumus,Z.H., Harding, S.T., Klepeis, J.L., Meyer, C.A., Schweiger, C. A., 1999, *Handbook of test problems in local and global optimization*, (Dordrecht: Kluwer Academic Publishers).

[7] Ierapetritou, M.G. and Pistikopoulos, E.N., 1995, Novel approach for optimal process design under uncertainty. *Computers and Chemical Engineering*, **19 (10)**,1089–1110.

[8] Yen, J. W. and Birge, J. R.,2006, A stochastic programming approach to the airline crew scheduling problem, *Transportation Science*, **40 (1)**, 3-14.

[9] Klein Haneveld, W.K. and van der Vlerk, M.H.,1999, Stochastic integer programming:General models and algorithms, *Annals of Operations Research*, **85 (1)**, 39-57.

[10] Wei, J. and Realff, J., 2004 , Sample average approximation methods for stochastic MINLPs, *Computers and Chemical Engineering*,**28 (3)**, 333–346.

[11] Norkin, V.I., Pflug, G.Ch., and Ruszcczyński, A., 1998, A branch and bound method for stochastic global optimization, *Mathematical Programming*, **83(3)**,425–450.

[12] Acevedo J. and Pistikopoulos E.N., 1996, A parametric MINLP algorithm for process synthesis problems under uncertainty, *Industrial and Engineering Chemistry Research*, **35 (1)**, 147–158.

[13] Acevedo, J. and Pistikopoulos, E.N., 1996, Computational studies of stochastic optimization algorithms for process synthesis under uncertainty, *Computers and Chemical Engineering*, **20**, s1–s6.

[14] Acevedo, J. and . Pistikopoulos,E.N, 1998, Stochastic optimization based algorithms for process synthesis under uncertainty, *Computers and Chemical Engineering*, **22 (4/5)**, 647–671.

[15] Ierapetrirou, M.G. and Acevedo, J. and Pistikopoulos, E.N., 1996, An optimization approach for process engineering problems under uncertainty, *Computers and Chemical Engineering*, **6-7**.

[16] Paules, E.G. and Floudas, C.A., 1992, Stochastic programming in process synthesis: a two-stage model with MINLP recourse for multiperiod heat-integrated distillation sequences, *Computers and Chemical Engineering*, **16**, 189–210.

[17] Shastri, Y. and Diwekar, U., 2000, An efficient algorithm for large scale stochastic nonlinear programming problems, *Computers and Chemical Engineering*, **30**, 864-877.

[18] Carøe, C.C and Schultz, R., 1999, Dual decomposition in stochastic integer programming, *Operations Research Letters*, **24**, 37–45.

[19] Schultz, R. 1995, On structure and stability in stochastic programs with random technology matrix and complete integer recourse, *Mathematical Programming*, **70**,73?-89.

[20] Ruszczyński, A., 1997, Decomposition methods in stochastic programming, *Mathematical Programming*, **79**, 333–353.

[21] Nasberg, M., Jonstern, K.O., Smeds, P.A., 1985, Variable splitting - a new Lagrangian relaxation approach to some mathematical programming problems, Technical Report, LITH-MATH-R-85-04, Linkoping University.

[22] Marshall, L. and Fisher, 1982, The Lagrangian relaxation method for solving integer programming problems, *Management Science*, **27(1)**,1–18.

[23] Guignard, M., Kim, S., 1987, Lagrangean decomposition: a model yielding stronger Lagrangean bound, *Mathematical Programming* , **39**, 215–228.

[24] Hemmecke, Schultz, R., 2001, Decomposition methods for two-stage stochastic integer programs. In: Online Optimization of Large Scale Systems, pp. 601-622. Springer, Berlin.

[25] Takriti S. and Birge,J.R., 2000, Lagrangean solution techniques and bounds for loosely coupled mixed-integer stochastic programs, *Operations Research*, **48 (1)**, 91–98.

[26] Römisch, W. and Schultz, R., Multi-stage stochastic integer programs: An introduction. In S.O. Krumke M. Grtchel and J. Rambau, (Ed.) *Online Optimization of Large Scale Systems*, pages 581–600. (Berlin: Springer) .

[27] Nedic, A., Bertsekas, D.P., 2002, Incremental subgradient methods for nondifferentiable optimization, *SIAM Journal on Optimization*, **12 (1)**, 109–138.

[28] Bertsekas, D. P.,1999, *Nonlinear Programming*, (2nd edition), Athena Scientific, Belmont.

[29] Borchers, B., Mitchell, J.E., 1994, An improved branch and bound algorithm for mixed integer nonlinear programs, *Computers and Operations Research*, **21**, 359–367.

[30] Leyffer, S., 2001, Integrating SQP and branch and bound for mixed integer nonlinear programming, *Computational Optimization and Applications*, **18**, 295-309.

[31] Dyckhoff, H., 1990, A typology of cutting and packing problems, *European Journal of Operations Research*, **1 (44)**, 145–159.

[32] Valé rio de Carvalho, J.M., 1998, Exact solution of one-dimensional cutting stock problems using column generation and branch and bound, *International Transactions in Operational Research*, **5(1)** , 35–44.

[33] Vance, F., 1998, Branch-and-price algorithms for the one-dimensional cutting stock problem. *Computational Optimization and Applications*, **9 (3)**, 212–228.

[34] Vanderbeck, F., 2000, Computational study of a column generation algorithm for bin packing and cutting stock problems, *Mathematical Programming*, **86**, 565–594.

[35] Chen, C., Hart, S., Tham, V., 1996, A simulated annealing heuristic for the one-dimensional cutting stock problem, *European Journal of Operations Research*, **93 (3)**, 522–535.

[36] Foerster, H. and Wascher, G., Simulated annealing for order spread minimization in sequencing cutting patterns, *European Journal of Operational Research*, **110 (2)**, 272–282.

[37] GAMS Ide Model Library. Example problem Trimloss. GAMS Development Corporation 1217 Potomac Street, NW Washington, DC, USA

[38] Lindo Api. User's Manual LINDO Systems, Inc. Chicago, Illinois

Received 19 Oct 07; revised 20 May 08; accepted 28 Aug 08