INFORMATICA, 2016, Vol. 27, No. 4, 799–818

DOI: http://dx.doi.org/10.15388/Informatica.2016.112

Solving Multistage Mixed Nonlinear Convex Stochastic Problems

Eugenio MIJANGOS

Department of Applied Mathematics and Statistics and Operations Research
P.O. Box 644, University of the Basque Country UPV/EHU, Spain
e-mail: eugenio.mijangos@ehu.eus

Received: November 2014; accepted: April 2016

Abstract. We present an algorithm to solve multistage stochastic convex problems, whose objective function and constraints are nonlinear. It is based on the twin-node-family concept involved in the Branch-and-Fix Coordination method. These problems have 0–1 mixed-integer and continuous variables in all the stages. The non-anticipativity constraints are satisfied by means of the twin-node-family strategy.

In this work to solve each nonlinear convex subproblem at each node we propose the solution of sequences of quadratic subproblems. Due to the convexity of the constraints we can approximate them by means of outer approximations. These methods have been implemented in C++ with the help of CPLEX 12.1, which only solves the quadratic approximations. The test problems have been randomly generated by using a C++ code developed by this author. Numerical experiments have been performed and its efficiency has been compared with that of a well-known code.

Key words: stochastic programming, convex programming, branch and fix coordination, mixed integer nonlinear programming, quadratic programming, outer approximation.

1. Introduction

Most decision problems involve uncertainty in some parameters. Stochastic programming is dedicated to developing methods for decision-taking under uncertainty over time in parameters of the optimization problems. In order to model the uncertainty over time, scenarios that approximate the future are used. In this field we have excellent theoretical books, see e.g. Birge and Louveaux (1997), Higle and Sen (1996), Kall and Wallace (1994), Prékopa (1995). There are also application books, see e.g. Uryasev and Pardalos (2001), Wallace and Ziemba (2005), Ziemba and Mulvey (1998).

In multistage programs, decisions on variables in each stage must be taken; e.g. first stage variables are chosen before knowing the realization of uncertain parameters in each scenario. Once decided on first stage and observed each realization of uncertain parameters, the second stage decision must be taken, and so on. For each stage there are variables associated with decisions that must be taken without anticipation of some of future problem data, that is, they take the same value in each scenario, which yields *non-anticipativity constraints*. If we consider a finite number of scenarios, a general multistage program can be expressed by means of the previous stage decision variables, which corresponds to a large programming problem suggested by Wets (1966) and known as *deterministic equivalent model* (**DEM**).

An important feature of this approach is that it deals with multistage programs with both continuous and binary variables at any stage. Elsewhere (Alonso-Ayuso *et al.*, 2003a; Escudero *et al.*, 2009), some approaches to address two-stage problems with only binary variables in the first stage and binary and continuous variables in the second stage were presented. Escudero *et al.* (2010) represented the multistage stochastic mixed 0–1 problem by a mixture of the compact and splitting variable representations of the **DEM** of the stochastic problem where each stage can have the binary and the continuous variables. It uses a specialization of the BFC scheme and of the twin-node-family (TNF) concept, which was introduced by Alonso-Ayuso *et al.* (2003b). This scheme is specifically designed for coordinating the node branching selection and pruning, and the 0–1 variable branching selection and fixing at each branch-and-fix (BF) tree. Also, in that paper the decomposition of the set of scenarios in clusters is suggested.

On the other hand, the mixed-integer nonlinear programs (MINLP) provide a powerful framework for mathematical modelling of problems including discrete and continuous decisions and nonlinearities. MINLP models have been developed and solved in various engineering areas, see among others (Floudas, 1995; Grossmann, 2002; Biegler and Grossmann, 2004). These problems have been traditionally solved with deterministic models, although the real systems are almost always uncertain. Stochastic programming is a natural way to address uncertainties in engineering problems (see Wallace and Ziemba, 2005). Mijangos (2013) puts forward an algorithm for solving two-stage stochastic problems with a quadratic objective function and linear constraints, which is used in Heredia *et al.* (2013) to solve electric market problems. It is based on the TNF concept involved in the BFC method. These problems have continuous and binary recourse variables in the first stage and only continuous variables in the second stage.

The present paper considers multistage stochastic problems with both continuous and binary variables at any stage. These problems have a nonlinear convex objective function, nonlinear convex constraints, and relatively complete recourse. The algorithm developed is based on BFC-MSMIP method of Escudero *et al.* (2010), but instead of solving an LP problem at each node of the BF trees, MINLP problems are solved and the coordinated tree-search and the iterative solution of the MINLP are interlaced. Thus the nonlinear part of our problem is solved whilst searching the tree. In this paper convex MINLP problems are solved via a sequence of quadratic programming approximations.

This paper is organized as follows: Section 2 defines the problem; Section 3 presents some basic theoretical results for the case where the subproblems are solved to optimality; Section 4 explains the solution of our problem using sequences of quadratic subproblems and outer approximations; and Section 5 puts forward the numerical tests. Finally, Section 6 concludes the paper.

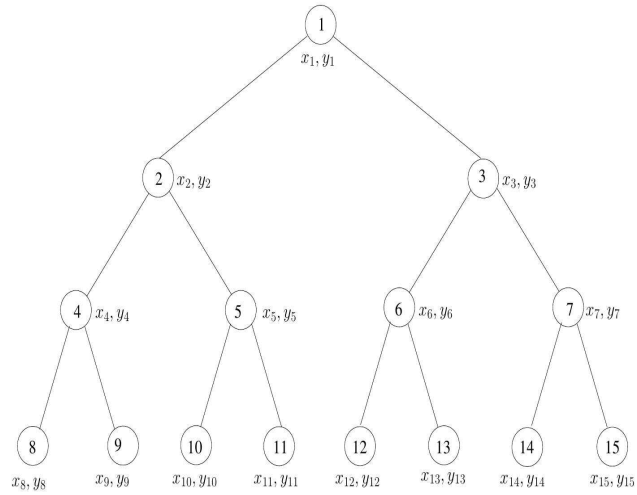

Fig. 1. Compact representation of a scenario tree.

2. Problem Definition

Consider the multistage mixed 0–1 nonlinearly-constrained nonlinear convex problem

$$\begin{aligned} \min\ & \sum_{t\in\mathcal{T}} c_t^T x_t + f_t(y_t) \\ \text{s.t.}\ & A_1 x_1 + h_1(y_1) \leqslant 0, \\ & A'_t x_{t-1} + A_t x_t + h_t(y_{t-1}, y_t) \leqslant 0, \quad t \in \mathcal{T} \setminus \{1\}, \\ & x_t \in \{0,1\}^{n_x}, \quad y_t \in \mathbb{R}^{+n_y}, \quad t \in \mathcal{T}, \end{aligned} \tag{1}$$

where $\mathcal{T}$ is the set of stages and for each $t \in \mathcal{T}$, $x_t \in \{0,1\}^{n_x}$ and $y_t \in \mathbb{R}^{+n_y}$ are t-stage variables, c_t is the coefficient vector for x_t in the objective function, $f_t \in \mathcal{C}^2$ is a nonlinear convex function, A'_t, A_t are constraint matrices, and $h_t \in \mathcal{C}$ are vector functions, whose components are convex nonlinear functions. Without loss of generality some constraints could be linear with lower and upper bounds.

We will introduce uncertainty in the parameters of the deterministic problem (1) by using a scenario analysis, see Fig. 1. It corresponds to the compact representation of the stochastic version, see **DEM** (2). Each node g represents a point in time where a decision can be made.

Once a decision is made in a stage, a number r of contingencies may occur (e.g. $r = 2$ for each stage t in Fig. 1). Information related to these contingencies is available at the beginning of each stage. $|\mathcal{T}|$ is the number of stages (e.g. $|\mathcal{T}| = 4$ in Fig. 1). At each stage, there are two kinds of decision variables: x, vector of 0–1 variables, and y, vector of continuous variables. The structure of this information is presented as a tree, where each root-to-leaf path represents one specific scenario ω and corresponds to one realization of all the uncertain parameters. Ω is the set of scenarios. Thus, in the tree of Fig. 1, we have $|\Omega| = 8$ root-to-leaf paths. Each node in the tree can be associated with a scenario group g, where $\mathcal{G}$ means the set of scenario groups and $\mathcal{G}_t$ the subset of scenario groups in stage t, such that $\mathcal{G} = \bigcup_{t\in\mathcal{T}} \mathcal{G}_t$. Two scenarios are in the same group in a given stage provided that they have the same realizations of the uncertain parameters up to the stage. Ω_g denotes the set of scenarios that belong to group g, for $g \in \mathcal{G}$. According to non-anticipativity principle, two scenarios should have the same value for the related variables with the time index up to the given stage.

Let us assume that the coefficient vector c, the function f, the constraint-matrix coefficients A and A', and the constraint functions h depend on the scenario groups. Then the compact representation of the mixed 0–1 **DEM** of the stochastic version with complete recourse of multistage problem can be given by

$$\begin{aligned} \min\ & \sum_{g\in\mathcal{G}} w_g\big(c_g^T x^g + f_g\big(y^g\big)\big) \\ \text{s.t.}\ & A_1 x^1 + h_1\big(y^1\big) \leqslant 0, \\ & A'_g x^{\pi(g)} + A_g x^g + h_g\big(y^{\pi(g)}, y^g\big) \leqslant 0, \quad g \in \mathcal{G}\setminus\{1\}, \\ & x^g \in \{0,1\}^{n_x}, \quad y^g \in \mathbb{R}^{+n_y}, \quad g \in \mathcal{G}, \end{aligned} \tag{2}$$

where w_g is the likelihood for g, such that $w_g = \sum_{\omega\in\Omega_g} w^\omega$, and $\pi(g)$ is the immediate predecessor node of node g. In addition, x^g, y^g are the x, y variables for the scenario group g. In this work, without loss of generality, we will consider that the dimensions of x^g and y^g are n_x and n_y, respectively, for each $g \in \mathcal{G}$.

As is showed by Escudero *et al.* (2010) the compact representation **DEM** can be written as a *splitting variable* representation, see Ruszczyński (1997); i.e. x^g and y^g are respectively replaced by x_t^ω, y_t^ω for the scenarios ω that belong to the same group Ω_g, for $g \in \mathcal{G}_t$ with $t \in \mathcal{T}^- = \mathcal{T} \setminus \{|\mathcal{T}|\}$. This gives rise to the following stochastic mixed-integer nonlinear problem:

$$\min \quad \sum_{\omega\in\Omega}\sum_{t\in\mathcal{T}} w^\omega\big(\big(c_t^\omega\big)^T x_t^\omega + f_t^\omega\big(y_t^\omega\big)\big) \tag{3}$$

$$\text{s.t.} \quad A_1^\omega x_1^\omega + h_1^\omega\big(y_1^\omega\big) \leqslant 0, \tag{4}$$

$$A_t^{'\omega} x_{t-1}^\omega + A_t^\omega x_t^\omega + h_t^\omega\big(y_{t-1}^\omega, y_t^\omega\big) \leqslant 0,\ \forall\omega\in\Omega,\ t\in\mathcal{T}\setminus\{1\}, \tag{5}$$

$$x_t^\omega - x_t^{\omega'} = 0, \quad \forall\omega\in\Omega_g,\ \omega' = \omega+1,\ \omega < |\Omega_g|,\ g\in\mathcal{G}_t,\ t\in\mathcal{T}^-, \tag{6}$$

$$y_t^\omega - y_t^{\omega'} = 0, \quad \forall\omega\in\Omega_g,\ \omega' = \omega+1,\ \omega < |\Omega_g|,\ g\in\mathcal{G}_t,\ t\in\mathcal{T}^-, \tag{7}$$

$$x_t^\omega \in \{0,1\}^{n_x}, \quad y_t^\omega \in \mathbb{R}^{+n_y}, \quad \forall\omega\in\Omega,\ t\in\mathcal{T}, \tag{8}$$

where (6) and (7) are the *non-anticipativity constraints* (NACs). Note that $f_t^\omega = f_g$, $c_t^\omega = c_g$ for $\omega \in \Omega_g$, $g \in \mathcal{G}_t$, $t \in \mathcal{T}$ and the same for the other parameters. The relaxation of these NACs in this model gives rise to $|\Omega|$ independent submodels:

$$\begin{aligned} \min\ & \sum_{t\in\mathcal{T}} w^\omega\big(\big(c_t^\omega\big)^T x_t^\omega + f_t^\omega\big(y_t^\omega\big)\big) \\ \text{s.t.}\ & A_1^\omega x_1^\omega + h_1^\omega\big(y_1^\omega\big) \leqslant 0, \\ & A_t^{'\omega} x_{t-1}^\omega + A_t^\omega x_t^\omega + h_t^\omega\big(y_{t-1}^\omega, y_t^\omega\big) \leqslant 0, \quad \forall t\in\mathcal{T}\setminus\{1\}, \\ & x_t^\omega \in \{0,1\}^{n_x}, \quad y_t^\omega \in \mathbb{R}^{+n_y},\ \forall t\in\mathcal{T}. \end{aligned} \tag{9}$$

These models are linked by the NACs.

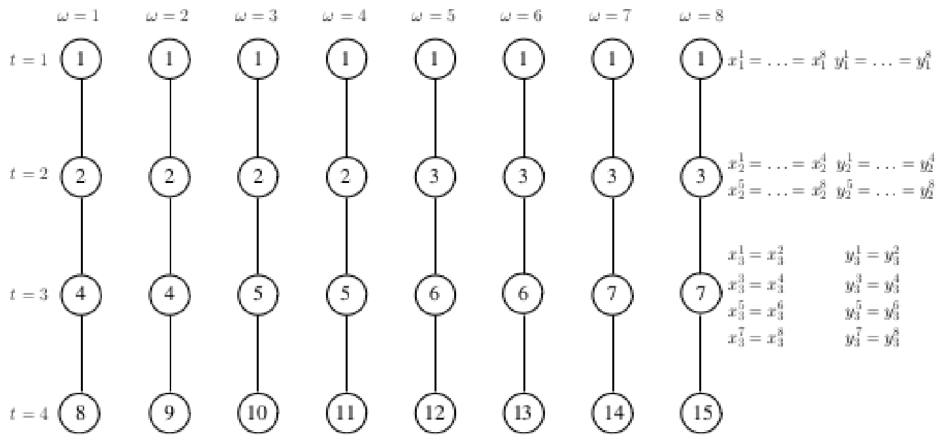

Fig. 2. Splitting variable representation of the scenario tree.

Figure 2 gives the tree of Fig. 1 by splitting the variables for the different root-to-leaf paths. On the right side of Fig. 2 for each stage except the last one we have the non-anticipativity constraints on the variables x_t^ω, y_t^ω for the scenarios ω that lie in the same scenario group Ω_g, for $g \in \mathcal{G}_t$. For example, for $t = 2$ stage and $\Omega_3 = \{5, 6, 7, 8\}$ scenario group, the equalities $x_2^5 = x_2^6 = x_2^7 = x_2^8$ and $y_2^5 = y_2^6 = y_2^7 = y_2^8$ must be satisfied.

3. Scenario Clusters

As suggested by Escudero *et al.* (2010), a scenario-cluster partitioning combines compact and splitting variable representations in the different stages of the problem, depending on the scenario cluster partition of choice.

Let q be the number of scenario clusters. This value is selected as a divisor of $|\Omega|$, then Ω^p gives the set of scenarios in cluster p with $|\Omega^p| = |\Omega|/q$, for $p = 1, \ldots, q$. For example, in Fig. 3 we have these set of scenarios: $\Omega^1 = \{1, 2\}, \ldots, \Omega^4 = \{7, 8\}$. In addition, let $\mathcal{G}^p \subseteq \mathcal{G}$ be the set of scenario groups for cluster p, such that $\Omega_g \cap \Omega^p \neq \emptyset$ means that $g \in \mathcal{G}^p$; e.g. in Fig. 3, $\mathcal{G}^1 = \{1, 2, 4, 8, 9\}, \ldots, \mathcal{G}^4 = \{1, 3, 7, 14, 15\}$.

Instead of the submodel (9) for $\omega \in \Omega$ we consider for each scenario cluster $p \in \{1, \ldots, q\}$ the compact representation

$$
\begin{aligned}
(\mathbf{SMINLP}^p)\ \min\ & \sum_{g \in \mathcal{G}^p} w_g \big(c_g^T x^g + f_g\big(y^g\big)\big) \\
\text{s.t.}\ \ & A_1 x^1 + h_1\big(y^1\big) \leqslant 0, \\
& A'_g x^{\pi(g)} + A_g x^g + h_g\big(y^{\pi(g)}, y^g\big) \leqslant 0, \quad g \in \mathcal{G}^p \setminus \{1\}, \\
& x^g \in \{0,1\}^{n_x}, \quad y^g \in \mathbb{R}^{+n_y}, \quad g \in \mathcal{G}^p,
\end{aligned}
\tag{10}
$$

where $w_g = \sum_{\omega \in \Omega_g \cap \Omega^p} w^\omega$.

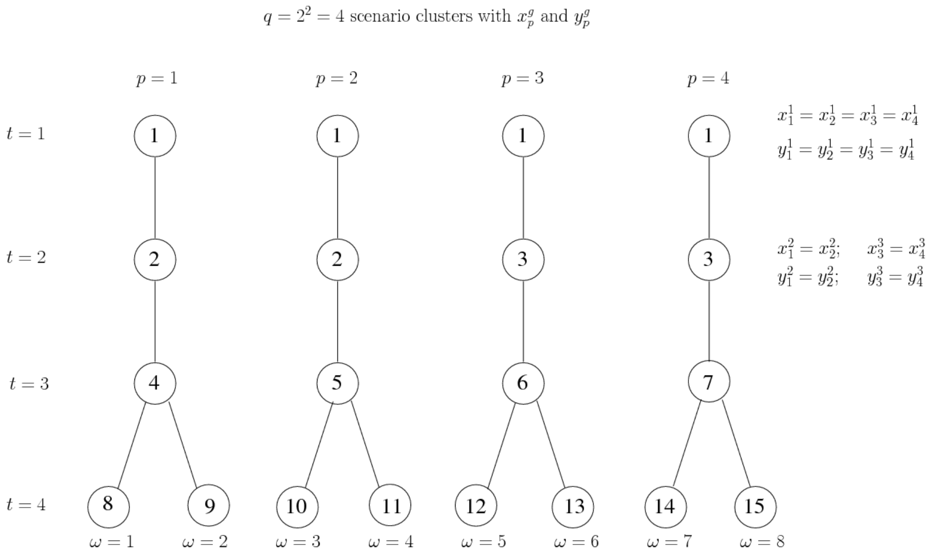

Fig. 3. Partition in $q = 4$ scenario clusters and NACs.

These q models are connected by the NACs

$$x_p^g - x_{p'}^g = 0 \quad (\text{NAC}_x),$$
$$y_p^g - y_{p'}^g = 0 \quad (\text{NAC}_y)$$

for all $g \in \mathcal{G}^p \cap \mathcal{G}^{p'}$, $p \in \{1, \ldots, q-1\}$ and $p' = p+1$, see Fig. 3. In future we will use x^g and y^g, with $g \in \mathcal{G}^p$, to denote the vectors x_p^g and y_p^g for each scenario cluster p.

Therefore, an equivalent model of (2) can be given in terms of the scenario-cluster models as

$$
\begin{aligned}
(\textbf{SMINLP}) \min \sum_{p=1}^{q} \sum_{g \in \mathcal{G}^p} & w_g\big(c_g^T x^g + f_g\big(y^g\big)\big) \\
\text{s.t.}\quad & A_1 x^1 + h_1\big(y^1\big) \leqslant 0, \\
& A'_g x^{\pi(g)} + A_g x^g + h_g\big(y^{\pi(g)}, y^g\big) \leqslant 0, \\
& g \in \mathcal{G}^p \setminus \{1\},\ p \in \{1, \ldots, q\}, \\
& x_p^g - x_{p'}^g = 0, \quad g \in \mathcal{G}^p \cap \mathcal{G}^{p'},\ p \in \{1, \ldots, q-1\},\ p' = p+1, \\
& y_p^g - y_{p'}^g = 0, \quad g \in \mathcal{G}^p \cap \mathcal{G}^{p'},\ p \in \{1, \ldots, q-1\},\ p' = p+1, \\
& x^g \in \{0,1\}^{n_x}, \quad y^g \in \mathbb{R}^{+n_y}, \quad g \in \mathcal{G}^p,\ p \in \{1, \ldots, q\}.
\end{aligned}
\tag{11}
$$

In general, without loss of generality, let us consider a symmetric and balanced scenario tree, with $|\Omega| = r^{|\mathcal{T}|-1}$, where r is the number of contingencies for each stage. Therefore, we can take the number of scenario clusters $q \in \{r, r^2, \ldots, r^{|\mathcal{T}|-1}\}$.

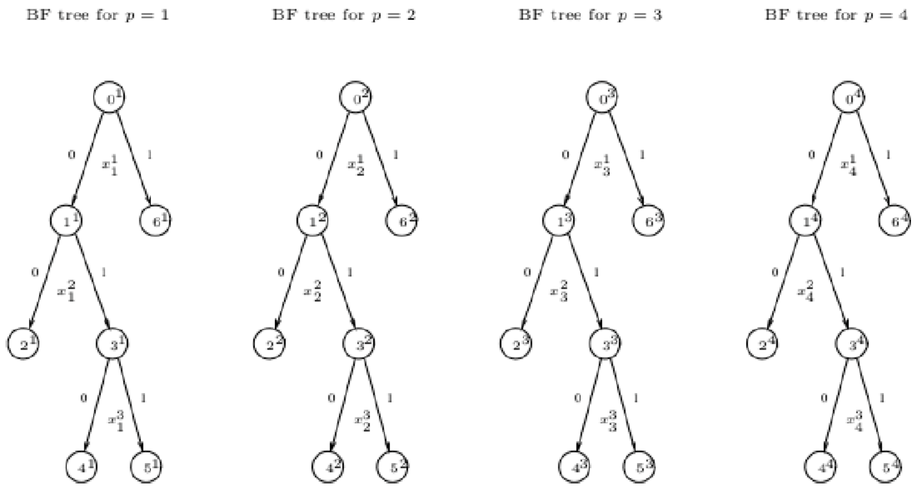

Fig. 4. Branch and fix trees for $p \in \{1, \ldots, q\}$ and $q = 4$ scenario clusters.

In sum, the model (11) is partitioned into q submodels (10), which are connected by the NACs. Each $\mathbf{SMINLP}^p$ submodel has $|\Omega^p| = |\Omega|/q$ scenarios.

4. Multistage BFC for Nonlinear Convexity

Given a partition in scenario clusters, the optimal solution for the models $\mathbf{SMINL}^p$, for $p = 1, \ldots, q$, is coordinately obtained by using the Branch-and-Fix Coordination scheme (BFC) based on the Twin-Node-Families concept (TNF), which was introduced by Alonso-Ayuso *et al.* (2003b). This strategy allows us a better coordination of the selection of the branching node and branching variable for each scenario cluster related BF tree, such that the relaxed NAC_x are satisfied when fixing the appropriate variables either to 1 or to 0. Also, it coordinates and reinforces the scenario cluster related BF node pruning and the variable fixing. BFC lies in the fact that it has as many branch and bound trees as the number of scenario clusters, coordinating the branching nodes (TNFs) and the branching variables (integer common variables) so that the related NAC_x are satisfied.

4.1. *Some Definitions*

Consider the *branch and fix (BF)* tree associated with the scenario cluster $p \in \{1, \ldots, q\}$ and generated by branching on/fixing at 0 or 1 the x^g variables, for $g \in \mathcal{G}^p$. Let $\mathcal{A}^p$ be the set of active nodes, which are nodes where there still are x-variables not fixed to 0 or to 1. Let also $\mathcal{I}$ be the set of indices of the binary variables x that the algorithm branches on. Denote $(x_p^g)_i$ the ith variable of the vector x_p^g.

In order to explain the concepts defined below we use Fig. 4, which corresponds to the scenario tree in Fig. 1, with the $q = r^2 = 4$ scenario clusters given in Fig. 3. The decision variables are x^1 in the first stage, and x^2, x^3 in the second stage. The four scenario clusters are connected by the NACs corresponding to the stages $t = 1$ and $t = 2$. $\mathcal{I}$ is the set of indices of the binary variables x for stages $t = 1$ and $t = 2$, and the branching order is x_p^1, x_p^2, x_p^3 for $p \in \{1, 2, 3, 4\}$.

We say that $(x_p^g)_i$ and $(x_{p'}^g)_i$ are *common variables* for scenario clusters p and p' when $i \in \mathcal{I}$, $g \in \mathcal{G}^p \cap \mathcal{G}^{p'}$, and p, p' are in Ω_g. For example, in Fig. 4 (without i indices), x_1^2 and x_2^2 are common variables, as in both scenario clusters 1 and 2 x^2 must be fixed to the same binary value. However, the variables x_1^2 and x_3^2 are not common, as in stage $t = 2$ we have $p = 1 \in \Omega_2$ and $p = 3 \in \Omega_3$, i.e. the scenario clusters are not in the same scenario group, see Fig. 3.

Let index $i \in \mathcal{I}$, scenario group $g \in \mathcal{G}^p \cap \mathcal{G}^{p'}$, and the scenario clusters $p, p' \in \{1, \dots, q\}$. Nodes $a \in \mathcal{A}^p$ and $a' \in \mathcal{A}^{p'}$ are *twin nodes* with respect to scenario group g when on the path from the root node to these nodes in each of the BF trees for p and p', the common variables $(x_p^g)_i$ and $(x_{p'}^g)_i$ have been branched on at the same binary value. For example, in Fig. 4, nodes 2^1 and 2^2 are twin nodes with respect to scenario group 2, as both have fixed the values of their common variables x_1^1, x_1^2 and x_2^1, x_2^2 to 0. However, 2^1 and 3^2 are not twin nodes, as their common variables x_1^2, x_2^2 have opposite values. 2^1 and 2^3 are not twin nodes, as in spite of the fact that variables x_1^2, x_3^2 have the same value, they are not common variables, since the scenario clusters are not in the same scenario group.

A *twin node family (TNF)* is a set of nodes such that any node is a twin node to all other nodes in the family. For example, in Fig. 4, the sets $\{0^1, 0^2, 0^3, 0^4\}$, $\{1^1, 1^2\}$, $\{1^3, 1^4\}$, $\{2^1, 2^2\}$, $\{2^3, 2^4\}$, $\{4^1, 4^2\}$ are TNFs.

A TNF is *candidate* when any of its common variables has not yet been fixed. For example, in Fig. 4, the TNF $\{1^1, 1^2\}$ is candidate, as its scenario cluster trees have in common variables x_1^2 and x_2^2 that have not yet been fixed. However, the TNF $\{1^1, 1^2, 1^3, 1^4\}$ is not a candidate, as the only common variable is x^1, but it has already been fixed.

A TNF integer set is a set of TNFs where all x variables have binary values and the NAC_x are satisfied. For example, in Fig. 4, the set $\{4^1, 4^2, 5^3, 5^4\}$ is a TNF integer set, as the integrality condition of x and the NAC_x are satisfied. However, the set $\{4^3, 5^4\}$ is not a TNF integer set, as the common variables x_3^3 and x_4^3 have taken different values, 0 and 1, respectively.

4.2. *Auxiliary Submodels*

Let $\mathcal{G}_t^*$ denote the cumulated set of scenario groups until stage t; i.e. $\mathcal{G}_t^* = \bigcup_{j=1}^t \mathcal{G}_j$ for $t \in \mathcal{T}$.

Consider $q = r^t$, for $t \in \{1, 2, \dots, |\mathcal{T}| - 1\}$, and a TNF where i of the x variables have been already branched on and fixed at for $i \in \mathcal{I}$ related to a group $g \in \mathcal{G}_t^*$. Let $n = |\mathcal{I}|$. We branch on the x^g variables, just for stages where there are explicitly NACs.

At each TNF, we compute the lower bound $\underline{Z}_i = \sum_{p=1}^q z_i^p$, where z_i^p is the optimal solution value of $\mathbf{MINLP}_i^p$, which denotes the mixed integer nonlinear problem obtained from $\mathbf{SMINLP}^p$ after the first i variables x^g have been fixed to 0 or 1, say, $\overline{x}_j^g$, such that $g \in \mathcal{G}^p \cap \mathcal{G}_t^*$, namely for $p \in \{1, \ldots, q\}$

$$
\begin{aligned}
(\mathbf{MINLP}_i^p) \quad z_i^p = \min \; & \sum_{g \in \mathcal{G}^p} w_g \big(c_g^T x^g + f_g(y^g)\big) \\
\text{s.t.} \; & A_1 x^1 + h_1(y^1) \leqslant 0, \\
& A'_g x^{\pi(g)} + A_g x^g + h_g\big(y^{\pi(g)}, y^g\big) \leqslant 0, \quad g \in \mathcal{G}^p \setminus \{1\}, \\
& x_j^g = \overline{x}_j^g, \quad 1 \leqslant j \leqslant i, \; g \in \mathcal{G}^p \cap \mathcal{G}_t^*, \\
& x_j^g \in [0,1], \quad i+1 \leqslant j \leqslant n, \; g \in \mathcal{G}^p \cap \mathcal{G}_t^*, \\
& x_j^g \in \{0,1\}, \quad n+1 \leqslant j \leqslant N_x, \; g \in \mathcal{G}^p \setminus \mathcal{G}_t^*, \\
& y^g \in \mathbb{R}^{+n_y}, \quad g \in \mathcal{G}^p,
\end{aligned}
\tag{12}
$$

where $N_x = n_x \cdot |\mathcal{G}|$. Note that in the computation of each lower bound $\underline{Z}_i$, q mixed 0–1 nonlinear convex subproblems must be solved, one per each scenario cluster p, with 0–1 variables in the stages $t+1$ until the last one.

$\underline{Z}_0$ denotes the lower bound associated with the root node, $i = 0$. Also, note that $\underline{Z}_0 \leqslant \underline{Z}_1 \leqslant \cdots \leqslant \underline{Z}_n$ and n is the number of nodes for branching and fixing the x^g variables, such that $g \in \mathcal{G}^p \cap \mathcal{G}_t^*$. If the optimal solution obtained in a TNF i satisfies the integrality of x and the NAC_x, two cases can happen with respect to the NAC_y:

1. If the NAC_y have been satisfied, the incumbent solution is updated and the TNF branch is pruned. If the set of active nodes is empty in the BF trees, that solution is the optimum.
2. Otherwise, to satisfy NAC_y we solve the submodel $\mathbf{MINLP}_i^{TNF}$ obtained by fixing in (2), **DEM**, the x^g-variables that satisfied integrality and NAC_x. If this problem is feasible, the incumbent solution is updated, and if the TNF cannot be pruned, we continue with the examination of the BF trees.

Submodel $\mathbf{MINLP}_i^{TNF}$ can be given as

$$
\begin{aligned}
Z_i^{TNF} = \min \; & \sum_{g \in \mathcal{G}} w_g \big(c_g^T x^g + f_g(y^g)\big) \\
\text{s.t.} \; & A_1 x^1 + h_1(y^1) \leqslant 0, \\
& A'_g x^{\pi(g)} + A_g x^g + h_g\big(y^{\pi(g)}, y^g\big) \leqslant 0, \quad g \in \mathcal{G} \setminus \{1\}, \\
& x_j^g = \overline{x}_j^g, \quad 1 \leqslant j \leqslant i, \; g \in \mathcal{G}_t^* \quad \text{(from branching/fixing)}, \\
& x_j^g = \overline{x}_j^g, \quad i+1 \leqslant j \leqslant n, \; g \in \mathcal{G}_t^* \quad (\text{from } \mathbf{MINLP}_i^p), \\
& x_j^g \in \{0,1\}, \quad n+1 \leqslant j \leqslant N_x, \; g \in \mathcal{G} \setminus \mathcal{G}_t^*, \\
& y^g \in \mathbb{R}^{+n_y}, \quad g \in \mathcal{G}.
\end{aligned}
\tag{13}
$$

Note that the solution of $\mathbf{MINLP}_i^{TNF}$ attached to a TNF integer set could be the incumbent solution, as it is a feasible solution of the original problem; so, Z_i^{TNF} is an upper bound of the solution value of the original problem. But it does not mean that it must be pruned unless $i = n$, as a better solution can still be obtained by following deep in the tree. For this reason, we use a very similar auxiliary submodel, $\mathbf{MINLP}_i^f$, which can be defined as

$$
\begin{aligned}
Z_i^f = \min\ & \sum_{g\in\mathcal{G}} w_g\big(c_g^T x^g + f_g(y^g)\big) \\
\text{s.t.}\ & A_1 x^1 + h_1(y^1) \leqslant 0, \\
& A'_g x^{\pi(g)} + A_g x^g + h_g\big(y^{\pi(g)}, y^g\big) \leqslant 0, \quad g \in \mathcal{G}\setminus\{1\}, \\
& x_j^g = \overline{x}_j^g, \quad 1 \leqslant j \leqslant i,\ g \in \mathcal{G}_t^* \quad \text{(from branching/fixing)}, \\
& x_j^g \in [0,1], \quad i+1 \leqslant j \leqslant n,\ g \in \mathcal{G}_t^*, \\
& x_j^g \in \{0,1\}, \quad n+1 \leqslant j \leqslant N_x,\ g \in \mathcal{G}\setminus\mathcal{G}_t^*, \\
& y^g \in \mathbb{R}^{+n_y}, \quad g \in \mathcal{G}.
\end{aligned}
\tag{14}
$$

Observe that, unlike $\mathbf{MINLP}_i^{TNF}$, in this submodel for $i+1 \leqslant j \leqslant n$ the variables x_j^g are relaxed in the interval $[0, 1]$. As a consequence, this model contributes strong lower bounds of the solution value of the descendant nodes from a given node. Therefore, the TNF can be pruned if $Z_i^f = Z_i^{TNF}$. Note that the solution of $\mathbf{MINLP}_i^f$ satisfies the NACs, but not necessarily the integrality for all x-variables. Hence, if Z_i^f is lower than Z_i^{TNF} and $\overline{Z}$, there are two possibilities. The first, the x-variables are integer, then we set $\overline{Z} := Z_i^f$ and the TNF is pruned. The second, there is any non-integer x-variable, then the branching follows deep to the $(i+1)$th node, since it is possible to find a better feasible solution.

5. Solution of SMINLP

In order to solve the original problem (2), equivalently **SMINLP**, we need to solve auxiliary subproblems. Since the objective function is convex and the constraints are convex, we propose to solve each $\mathbf{MINLP}^p$ in each node of the BF tree (in each TNF), and $\mathbf{MINLP}^{TNF}$ and $\mathbf{MINLP}^f$ subproblems by solving sequences of mixed integer quadratic problems (**MIQP**).

Since the components of h_g are convex, we propose to approximate each convex constraint by using *linear outer approximations* (see Duran and Grossmann, 1986), i.e. by replacing each constraint $j_g = 1, \ldots, m$, for $g \in \mathcal{G}\setminus\{1\}$, with the linear constraint

$$
\big(a'_{j_g}\big)^T x^{\pi(g)} + a_{j_g}^T x^g + h_{j_g}\big(y_k^{\pi(g)}, y_k^g\big) + \nabla h_{j_g}\big(y_k^{\pi(g)}, y_k^g\big)^T \begin{pmatrix} y^{\pi(g)} - y_k^{\pi(g)} \\ y^g - y_k^g \end{pmatrix} \leqslant 0,
$$

where $(y_k^{\pi(g)}, y_k^g)$ is the solution in the previous iteration, a_{j_g} is the j-th row of A_g, and h_{j_g} function is the j-th component of h_g. We follow a similar approach for $g = 1$.

Each $\mathbf{MINLP}_i^{TNF}$ is solved by sequentially solving the mixed integer quadratic problem $\mathbf{MIQP}_k^{TNF}$ given by

$$
\begin{aligned}
\Theta_k^{TNF} = \min\ & \sum_{g\in\mathcal{G}} w_g\left(c_g^T x^g + q_k^g(y^g)\right)\\
\text{s.t.}\ & a_{j_1}^T x^1 + h_{j_1}\left(y_k^1\right) + \nabla h_{j_1}\left(y_k^1\right)^T\left(y^1 - y_k^1\right) \leqslant 0, \quad j_1 = 1,\dots,m,\\
& \left(a'_{j_g}\right)^T x^{\pi(g)} + a_{j_g}^T x^g + h_{j_g}\left(y_k^{\pi(g)}, y_k^g\right)\\
& + \nabla h_{j_g}(y_k^{\pi(g)}, y_k^g)^T \begin{pmatrix} y^{\pi(g)} - y_k^{\pi(g)} \\ y^g - y_k^g \end{pmatrix} \leqslant 0,\\
& j_g = 1,\dots,m,\ g \in \mathcal{G}\setminus\{1\},\\
& x_j^g = \overline{x}_j^g, \quad 1 \leqslant j \leqslant i,\ g \in \mathcal{G}_t^* \quad \text{(from branching/fixing)},\\
& x_j^g = \overline{x}_j^g, \quad i+1 \leqslant j \leqslant n,\ g \in \mathcal{G}_t^* \quad (\text{from } \mathbf{MINLP}_i^p),\\
& x_j^g \in \{0,1\}, \quad n+1 \leqslant j \leqslant N_x,\ g \in \mathcal{G}\setminus\mathcal{G}_t^*,\\
& y^g \in \mathbb{R}^{+n_y}, \quad g \in \mathcal{G},
\end{aligned}
$$

where

$$
\begin{aligned}
q_k^g\left(y^g\right) = {} & f_g\left(y_k^g\right) + \nabla f_g\left(y_k^g\right)^T\left(y^g - y_k^g\right)\\
& + (1/2)\left(y^g - y_k^g\right)^T \nabla^2 f_g\left(y_k^g\right)\left(y^g - y_k^g\right)
\end{aligned}
\tag{15}
$$

is the Taylor polynomial of $f_g(y^g)$.

The vector y_k^g is updated $\overline{k}$ times at most and each y_k^g is the solution of the previous quadratic approximation, i.e. that corresponding to $q_{k-1}^g(y^g)$. The initial y_1^g is the estimate solution from the previous node. Thus, we solve $\overline{k}$ $\mathbf{MIQP}_k^{TNF}$ approximations at most. This sequence is stopped when for a given τ it holds

$$
\left|\Theta_k^{TNF} - \Theta_{k-1}^{TNF}\right| < \tau \cdot \left|\Theta_{k-1}^{TNF}\right|, \quad k \leqslant \overline{k},
$$

then, we set $Z_i^{TNF} := \Theta_k^{TNF}$.

In the case of $\mathbf{MINLP}_i^f$ we use a similar method to obtain Z_i^f. In addition, since we want to use its solution value like a strong lower bound, and f_g functions are convex, the following *objective cut* is included in the $\mathbf{MIQP}_k^f$ subproblems:

$$
\sum_{g\in\mathcal{G}} w_g\left(c_g^T x^g + f_g\left(y_k^g\right) + \nabla f_g\left(y_k^g\right)^T\left(y^g - y_k^g\right)\right) \leqslant \overline{Z},
\tag{16}
$$

where $\overline{Z} = \min\{\overline{Z}, Z_i^{TNF}\}$. Therefore, the $\mathbf{MIQP}_k^f$ submodel has the following formulation:

$$
\begin{aligned}
\min\ & \sum_{g\in\mathcal{G}} w_g\big(c_g^T x^g + q_k^g(y^g)\big)\\
\text{s.t.}\ & \sum_{g\in\mathcal{G}} w_g\big(c_g^T x^g + f_g\big(y_k^g\big) + \nabla f_g\big(y_k^g\big)^T\big(y^g - y_k^g\big)\big) \leqslant \overline{Z},\\
& a_{j_1}^T x^1 + h_{j_1}\big(y_k^1\big) + \nabla h_{j_1}\big(y_k^1\big)^T\big(y^1 - y_k^1\big) \leqslant 0, \quad j_1 = 1,\dots,m,\\
& \big(a'_{j_g}\big)^T x^{\pi(g)} + a_{j_g}^T x^g + h_{j_g}\big(y_k^{\pi(g)}, y_k^g\big)\\
& + \nabla h_{j_g}\big(y_k^{\pi(g)}, y_k^g\big)^T \begin{pmatrix} y^{\pi(g)} - y_k^{\pi(g)}\\ y^g - y_k^g \end{pmatrix} \leqslant 0,\\
& j_g = 1,\dots,m,\ g\in\mathcal{G}\setminus\{1\},\\
& x_j^g = \overline{x}_j^g, \quad 1\leqslant j\leqslant i,\ g\in\mathcal{G}_t^* \quad \text{(from branching/fixing)},\\
& x_j^g \in [0,1], \quad i+1\leqslant j\leqslant n,\ g\in\mathcal{G}_t^*,\\
& x_j^g \in \{0,1\}, \quad n+1\leqslant j\leqslant N_x,\ g\in\mathcal{G}\setminus\mathcal{G}_t^*,\\
& y^g \in \mathbb{R}^{+n_y}, \quad g\in\mathcal{G}.
\end{aligned}
\tag{17}
$$

If $\mathbf{MIQP}_k^f$ is infeasible, the node is pruned.

For the solution of $\mathbf{MINLP}_i^p$ subproblems in each node i of the TNF we solve the following mixed integer quadratic problem $\mathbf{MIQP}_i^p$:

$$
\begin{aligned}
z_i^p := \min\ & \sum_{g\in\mathcal{G}^p} w_g\big(c_g^T x^g + q_k^g(y^g)\big)\\
\text{s.t.}\ & a_{j_1}^T x^1 + h_{j_1}\big(y_k^1\big) + \nabla h_{j_1}\big(y_k^1\big)^T\big(y^1 - y_k^1\big) \leqslant 0, \quad j_1 = 1,\dots,m,\\
& \big(a'_{j_g}\big)^T x^{\pi(g)} + a_{j_g}^T x^g + h_{j_g}\big(y_k^{\pi(g)}, y_k^g\big)\\
& + \nabla h_{j_g}\big(y_k^{\pi(g)}, y_k^g\big)^T \begin{pmatrix} y^{\pi(g)} - y_k^{\pi(g)}\\ y^g - y_k^g \end{pmatrix} \leqslant 0,\\
& j_g = 1,\dots,m,\ g\in\mathcal{G}^p\setminus\{1\},\\
& x_j^g = \overline{x}_j^g, \quad 1\leqslant j\leqslant i,\ g\in\mathcal{G}^p\cap\mathcal{G}_t^*,\\
& x_j^g \in [0,1], \quad i+1\leqslant j\leqslant n,\ g\in\mathcal{G}^p\cap\mathcal{G}_t^*,\\
& x_j^g \in \{0,1\}, \quad n+1\leqslant j\leqslant N_x,\ g\in\mathcal{G}^p\setminus\mathcal{G}_t^*,\\
& y^g \in \mathbb{R}^{+n_y}, \quad g\in\mathcal{G}^p,
\end{aligned}
$$

where $q_k^g(y^g)$ is defined by (15). We only solve one quadratic approximation, i.e. $\overline{k} = 1$, since if for $k = 1$ and any p scenario cluster $\mathbf{MIQP}_i^p$ is infeasible, $\mathbf{MINLP}_i^p$ is infeasible (as its constraints are convex) and then the node of BFC is pruned. If it is feasible, due to the successive iterations in the previous nodes, the current values of (x_1^g, y_1^g) have been "polished", hence they are a good reference for the quadratic approximation. As a consequence, the optimal solution for $\mathbf{MINLP}_i^p$ subproblems will not be very different from the optimal solution for $\mathbf{MIQP}_i^p$ subproblems with $\overline{k} = 1$, which reduces drastically the computational effort required by this method.

For the solution of the $\mathbf{MIQP}^{TNF}$ and $\mathbf{MIQP}^f$ subproblems, by default, $\overline{k} = 20$ and $\tau = 10^{-6}$. These values have been heuristically chosen.

For a given $t \in \{1, \dots, |\mathcal{T}| - 1\}$ and $q = r^t$, an *initial lower bound* on the solution value of the original problem is obtained at the root node ($i = 0$) by

$$\underline{Z}_0 = \sum_{p=1}^{q} z_{00}^p,$$

where z_{00}^p is obtained by solving the following subproblem for each scenario cluster $p \in \{1, \dots, q\}$

$$\begin{aligned} z_{00}^p = \min\ & \sum_{g \in \mathcal{G}^p} w_g\big(c_g^T x^g + f_g\big(y^g\big)\big) \\ \text{s.t.}\ & A_1 x^1 + h_1\big(y^1\big) \leqslant 0, \\ & A'_g x^{\pi(g)} + A_g x^g + h_g\big(y^{\pi(g)}, y^g\big) \leqslant 0, \quad g \in \mathcal{G}^p \setminus \{1\}, \\ & x^g \in \{0, 1\}^{n_x}, \quad g \in \mathcal{G}^p, \\ & y^g \in \mathbb{R}^{+n_y}, \quad g \in \mathcal{G}^p. \end{aligned} \tag{18}$$

Note that the x-variables take 0–1 values in all stages.

Let the initial branching parameter σ_i be the most repeated 0–1 value of the x_i variable, for the clusters where the NAC_x must be satisfied, in the solution of (18) for $p \in \{1, \dots, q\}$, for each $i \in \{1, \dots, n\}$ ($n = |\mathcal{I}|$). If there is the same number of 0 and 1, we set $\sigma_i = 0$. Later we branch on the other value, which is denoted by $\overline{\sigma}_i$. We arrange the branching variables according to the initial natural ordering.

Note that in the algorithm given by Fig. 5 the upper index g, with $g \in \mathcal{G}^p$, denotes the set of common variables, x_i^g of cluster p, that must be fixed to the same 0–1 value. Likewise, the subindex i denotes the index in $\mathcal{I}$, i.e. the corresponding set over which the algorithm proceeds by branching on. Moreover, $\mathcal{I}$ is the subset of variables until the fixed stage t such that $q = r^t$ and IC means integrality constraints.

6. Numerical Results

This method has been implemented with the help of CPLEX 12.1 to solve the quadratic subproblems $\mathbf{MIQP}^p$ in each node of the BF tree, for each $p \in \{1, \dots, q\}$, and to solve the quadratic subproblems $\mathbf{MIQP}^{TNF}$ and $\mathbf{MIQP}^f$.

MS-NLBFC code is a C++ implementation of this method with the coordination of x in the twin node families of the BF trees for clusters $p \in \{1, \dots, q\}$.

Three convex functions have been used as $f(y)$ for $y_i \geqslant 0$, $\forall i$:

$$p(y) = \sum_{i=1}^{n_y} \big(y_i^3 + c_i y_i\big),$$

$$e(y) = \sum_{i=1}^{n_y} \big(\exp(y_i) + c_i y_i\big), \quad \text{and} \quad l(y) = \sum_{i=1}^{n_y} \big(-\log(y_i + 1) + c_i y_i\big).$$

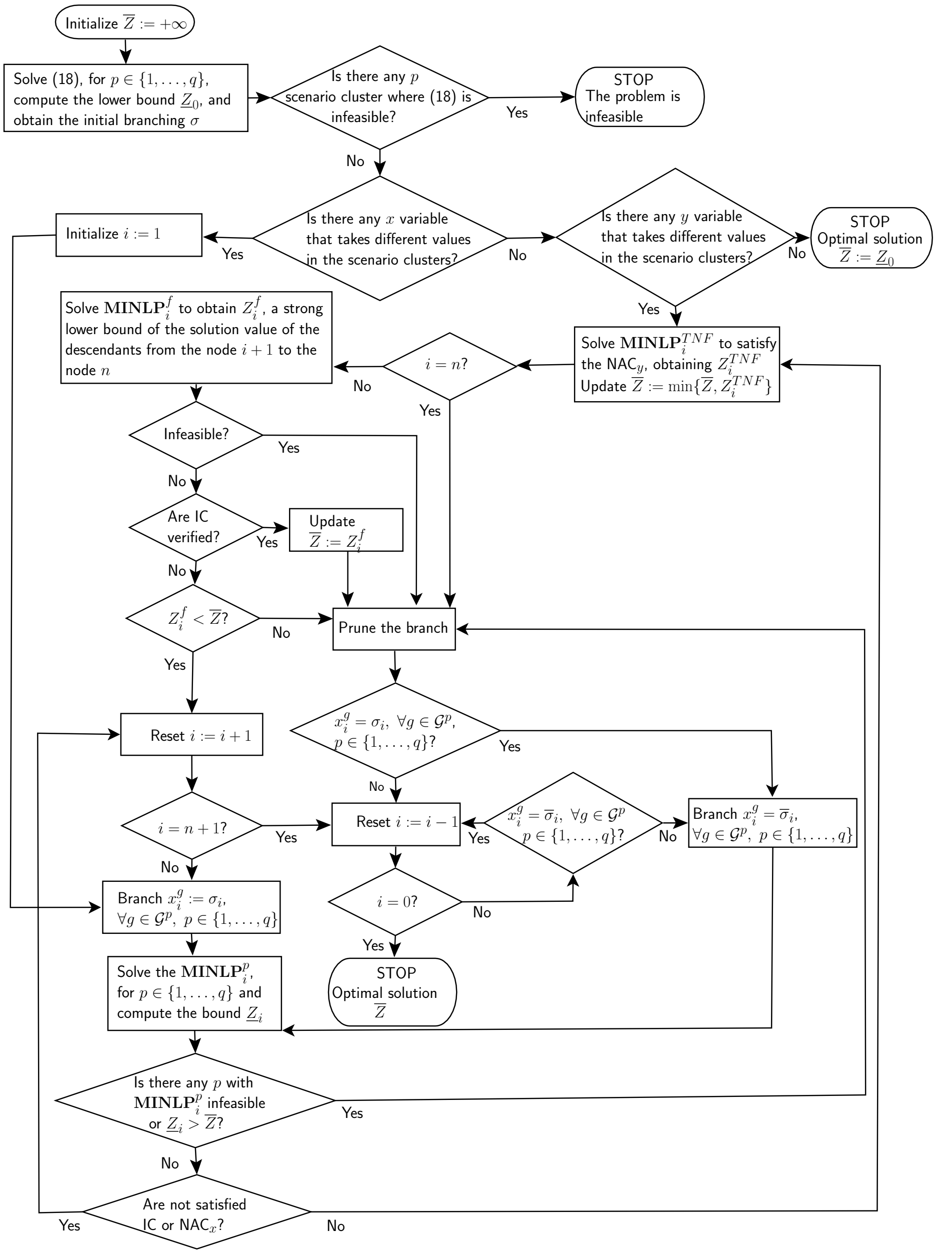

Fig. 5. Flowchart of the algorithm.

The constraint functions $h(y)$ are composed of a linear combination of the terms y_i^2 with positive coefficients, which are randomly chosen. The first-stage constraints are two-sided inequality linear constraints.

Table 1
Dimensions of the problems.

Test	Scenario model			Group dim.			DEM dim.				
	Tree	$\|\Omega\|$	$\|\mathcal{G}\|$	n_x	n_y	m	#bin	#nlvar	#lc	#nlc	dens%
T1	2^3	8	15	5	15	20	75	225	20	280	60
T2	2^3	8	15	10	30	42	150	450	42	588	48
T3	3^3	27	40	4	4	4	160	160	4	156	75
T4	3^3	27	40	8	8	8	320	320	8	312	75
T5	3^3	27	40	10	10	10	400	400	10	390	30
T6	3^3	27	40	15	15	15	600	600	15	585	54
T7	3^3	27	40	10	25	20	400	1000	20	780	60
T8	3^3	27	40	10	30	22	400	1200	22	858	59
T9	4^3	64	85	5	20	15	425	1700	15	1275	40
T10	5^3	125	156	5	12	12	780	1872	12	1860	33
T11	5^3	125	156	15	15	15	2340	2340	15	2325	47
T12	6^3	216	259	5	20	15	1295	5180	15	3870	47
T13	8^3	512	585	5	5	5	2925	2925	5	2920	40
T14	8^3	512	585	3	12	12	1755	7020	12	7008	25

Table 1 presents the dimensions of the 14 test problems according to the scenario model, the dimensions of each scenario group, and the dimensions of **DEM** model, see (2). In regard to "Scenario model" the headings are as follows: "Tree" indicates the kind of tree; $|\Omega|$ the number of scenarios; and $|\mathcal{G}|$ the number of scenario groups. "Group dim." has these headings: n_x, number of binary variables in each group; n_y, number of continuous variables in each group; and m, number of constraints in each group. Finally, under "DEM dim." there are these columns: #bin, number of binary variables x; #nlvar, number of nonlinear variables y; #lc, number of linear constraints; #nlc, number of nonlinear constraints; and "dens%", density of the constraints.

Numerical experiments have been performed on HP **Compaq** with Intel Core 2 Quad Q9550 2.83GHz 4 CPU under Linux and 4 GB of RAM. AMPL modelling system has been used as an interface with the solver BONMIN (Basic Open-Source Nonlinear Mixed INteger programming), which is an experimental open-source C++ code for solving general MINLP by (Bonami *et al.*, 2008) (COIN-OR).

Tables 2, 3, and 4 show, for $q = r$ clusters, the results of the computational experiments for our tests with polynomial function $p(y)$, with exponential function $e(y)$, and with logarithmic function $l(y)$, respectively. In these tables the headings are as follows: #nF, number of mixed 0–1 problems (13) solved; #n, number of TNF branches for the set of BF trees; f_M, solution value of the original problem obtained by MS-NLBFC; t_M, computing time required by MS-NLBFC to solve each problem in CPU-seconds; f_B, solution value of the original problem obtained by BONMIN; t_B, computing time required by BONMIN to solve each problem in CPU-seconds. Symbol "–" means that BONMIN finishes the execution with the message "The LP relaxation is infeasible or too expansive".

As can be seen in Table 2 for the polynomial function $p(y)$, with the exception of T6, in the rest of the tests the efficiency of MS-NLBFC has been higher than that of BONMIN, and especially in the biggest instances T7-T9 and T11-T14. BONMIN does not find a solution for T10. In Table 3 for the exponential function $e(y)$, in all the instances MS-

Table 2
Numerical results for $p(y)$.

Test	#nF	#n	f_M	t_M	f_B	t_B
T1	0	0	3709.6585	0.9	3709.6581	4.1
T2	2	32	89.242415	4.3	89.242408	13.7
T3	2	12	999.91318	0.6	999.91305	1.1
T4	4	34	3155.1924	3.4	3155.1922	5.3
T5	0	0	3218.9060	0.8	3218.9058	1.4
T6	3	78	5051.4940	13.9	5051.4936	9.6
T7	0	0	6485.5299	2.2	6486.0032	11.6
T8	0	0	5978.8555	2.1	5978.8548	14.3
T9	0	0	1416.3723	2.1	1416.3718	29.5
T10	0	0	3338.5685	2.8	–	–
T11	0	0	7313.7601	2.3	7313.7599	95.8
T12	0	0	64936.258	3.5	64936.256	1253.2
T13	1	14	1718.9321	6.6	1718.9320	803.5
T14	0	0	24933.784	5.6	24933.783	97.0

Table 3
Numerical results for $e(y)$.

Test	#nF	#n	f_M	t_M	f_B	t_B
T1	0	0	3775.5283	0.9	3775.5280	2.1
T2	0	0	220.39991	2.8	220.39990	17.4
T3	2	12	1016.8698	0.6	1016.8697	1.2
T4	2	30	3190.6568	2.6	3190.6566	4.6
T5	0	0	3254.3535	0.9	3254.3532	1.7
T6	3	78	5110.9118	15.6	5110.9115	199.6
T7	0	0	6585.8601	1.6	6585.8600	5.9
T8	0	0	6103.6315	2.2	6103.6307	10.0
T9	0	0	1500.2018	2.2	1500.2013	64.9
T10	0	0	3390.2928	3.0	–	–
T11	0	0	11714.366	3	11714.366	7.6
T12	0	0	81307.035	6.1	81307.033	1155.8
T13	1	14	1741.1116	6.6	1741.1115	6113.7
T14	0	0	30412.109	7.4	30412.108	300.4

NLBFC gives a lower time than BONMIN, except for T10, where this last does not converge. In the biggest instances T12-T14 the times obtained by MS-NLBFC in comparison with those of BONMIN are especially significant. Finally, in Table 4 for the logarithmic function $l(y)$, in the tests T9, T11 and T12 BONMIN does not find a solution. However, MS-NLBFC converges in all the instances and with a high efficiency with regard to that of BONMIN.

The relative difference between the values of the objective function for both solvers at the optimal is about 10^{-8} in all the instances for $p(y)$ and $e(y)$ functions. In the case of the function $l(y)$ in the instances T2, T4, T7, T10, and T13 MS-NLBFC gives a clearly lower value of the objective function in the solution than BONMIN, especially in T10 and T13, in these tests MS-NLBFC's solution is about 4% better (lower) than that computed by BONMIN.

Table 4
Numerical results for $l(y)$.

Test	#nF	#n	f_M	t_M	f_B	t_B
T1	0	0	3698.8769	0.9	3698.8766	2.1
T2	2	34	59.519263	7.1	59.580983	89.6
T3	2	12	994.87526	0.6	994.87512	0.8
T4	4	34	3149.3124	3.5	3158.6708	16.3
T5	0	0	3185.3608	1.1	3185.3606	11.4
T6	3	78	5026.4560	14.2	5026.4557	56.7
T7	0	0	6443.2200	2.8	6447.1188	134.0
T8	0	0	5956.5374	2.2	5956.5367	54.4
T9	0	0	1411.2299	2.3	–	–
T10	0	0	3329.0272	2.9	3473.4372	207.3
T11	14	116	3047.4103	119.5	–	–
T12	7	24	51847.349	100.3	–	–
T13	1	14	1716.5050	7.5	1780.3682	621.2
T14	0	0	22161.360	5.1	22161.337	954.5

Table 5
Experiments for q values (#nF; #n; t_M).

Test	$q = r^1$	$q = r^2$	$q = r^3$
T2	(2; 35; 7.1)	(2; 35; 10.7)	(2; 35; 20.6)
T3	(2; 13; 0.6)	(2; 13; 1.3)	(2; 13; 3.9)
T4	(4; 35; 3.5)	(4; 35; 6.8)	(4; 35; 23.8)
T6	(3; 79; 14.2)	(3; 79; 37.5)	(3; 79; 171.0)
T13	(1; 15; 7.5)	(1; 15; 40.2)	(2; 17; 842.4)

Table 6
Experiments using objective cuts (#**MIQP**^{f}/t_M).

Test	$q = r^1$		$q = r^2$		$q = r^3$	
	!OC	OC	!OC	OC	!OC	OC
T4	44/5.0	25/3.5	44/9.1	25/6.8	44/28.0	25/23.8
T6	22/17.1	3/14.2	42/48.3	4/37.5	42/201.0	4/171.0

Table 5 shows the performance of MS-NLBFC in some tests for different values of q using $l(y)$ in the objective function. As could be expected the bigger the number of clusters, the higher the computing time is. This is owing to the larger amount of subproblems (12) that must be solved, despite the smaller size of the subproblems.

Finally, the use of objective cuts (16) reduces the number of **MIQP**^{f} subproblems (17) that must be solved. In Table 6, for the instances T4 and T6 with $l(x)$ function, for each kind of q we have two columns: !OC, not using objective cuts; OC, using objective cuts. In the instance T6, for $q = r$, if we use objective cuts, instead of solving 22 **MIQP**^{f} problems only 3 are solved and, as a consequence, the total run-time is also reduced, from 17.1 seconds to 14.2.

7. Conclusions

An algorithm based on the Branch-and-Fix Coordination method has been designed to solve multistage stochastic mixed integer nonlinear convex problems with convex objective function and constraints. The uncertainty of these problems appears in the coefficients of the objective function and of the constraints.

The algorithm approximates the objective function by a sequence of quadratic Taylor polynomials. The convex constraints are approximated by means of linear outer approximations. When solving the submodels related to fractional TNFs objective cuts are involved in order to reduce the computational effort when the subproblem has an optimal solution bigger than that obtained for the integer TNF submodel or than the current incumbent solution.

It has been implemented in C++ with the help of CPLEX library to solve only the quadratic subproblems. This algorithm has been tested with a set of small- and medium-sized instances. The preliminary numerical results show that this experimental code is able to efficiently solve this kind of problems.

Some important topics for further investigation are the following: the Benders Decomposition for solving the submodels related to integer and fractional TNFs in the MS-NLBFC algorithm, the parallelization of the scenario-cluster related submodels solving, and the application of this code to solve real-world problems.

Acknowledgements. This research has been partially supported by the grant MTM2013-48462-C2-1-R of the Spanish research program. The author would like to express appreciations to the anonymous reviewers for valuable comments and suggestions that helped to improve the paper.

References

Alonso-Ayuso, A., Escudero, L.F., Garín, A., Ortuño, M., Pérez, G. (2003a). An approach for strategic supply chain planning based on stochastic 0–1 programming. *Journal of Global Optimization*, 26, 97–124.

Alonso-Ayuso, A., Escudero, L.F., Ortuño, M. (2003b). BFC, a branch-and-fix coordination algorithmic framework for solving some types of stochastic pure and mixed 0–1 programs. *European Journal of Operational Research*, 151, 503–519.

Biegler, L.T., Grossmann, I.E. (2004). Retrospective on optimization. *Computers and Chemical Engineering*, 28, 1169–1192.

Birge, J., Louveaux, F. (1997). *Introduction to Stochastic Programming*. Springer.

Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wätcher, A. (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2), 186–204.

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

Escudero, L.F., Garín, M., Merino, M., Pérez, G. (2009). A general algorithm for solving two-stage stochastic mixed 0–1 first-stage problems. *Computers and Operations Research*, 36, 2590–2600.

Escudero, L.F., Garín, M., Merino, M., Pérez, G. (2010). On BFC-MSMIP strategies for scenario cluster partitioning, and twin node family branching selection and bounding for multistage stochastic mixed integer programming. *Computers and Operations Research*, 37, 738–753.

Floudas, C.A. (1995). *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Oxford University Press, London.

Grossmann, I.E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization Engineering*, 28, 1169–1192.

Heredia, F.J., Corchero, C., Mijangos, E. (2013). Solving electric market quadratic problems by branch and fix coordination methods. In: Hömberg, D., Tröltzsch, F. (Eds.), *System Modeling and Optimization, CSMO 2011*, IFIP AICT, Vol. 391. Springer, pp. 511–520.

Higle, J.L., Sen, S. (1996). *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Programming*. Kluwer Academic Publishers, Dordrecht.

Kall, P., Wallace, S.W. (1994). *Stochastic Programming*. Wiley, Chichester.

Mijangos, E. (2013). An algorithm for two-stage quadratic problems. In: Hömberg, D., Tröltzsch, F. (Eds.), *System Modeling and Optimization, CSMO 2011*, IFIP AICT, Vol. 391. Springer, pp. 177–187.

Prékopa, A. (1995). *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht.

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

Uryasev, S., Pardalos, P. (2001). *Stochastic Optimization: Algorithms and Applications*. Kluwer Academic Publishers, Dordrecht.

Wallace, S.W., Ziemba, W.T. (2005). *Applications of Stochastic Programming. MPS-SIAM Series on Optimization*.

Wets, R.J.B. (1966). Programming under uncertainty: the equivalent convex program. *SIAM Journal on Applied Mathematics*, 14, 89–105.

Ziemba, W.T., Mulvey, J. (1998). *Worldwide Asset and Liability Modeling*. Cambridge University Press, Cambridge.

Eugenio Mijangos is an Associate Professor of Mathematics in the Department of Applied Mathematics and Statistics and Operations Research at University of the Basque Country (UPV/EHU), Leioa, Bizkaia, Spain. He received his degree in Science from UPV/EHU and the PhD in Science from Universitat Politècnica de Catalunya, Barcelona, Spain. His main research interests are continuous optimization, stochastic programming and liberalized electricity market.

Daugiaetapio mišraus netiesinio stochastinio iškilo programavimo uždavinių sprendimas

Eugenio MIJANGOS

Pateikiamas algoritmas daugiaetapio stochastinio iškilo programavimo uždaviniams spręsti, su netiesinėmis tikslo ir ribojimų funkcijomis. Algoritmas grindžiamas dvigubų mazgų koncepcija, įtraukta į koordinačių Plėtimosi ir Fiksavimo metodą. Neanticipatyviškumo sąlygos yra tenkinamos taikant dvigubų mazgų strategiją. Šiame darbe siūloma spręsti pagalbinius uždavinius pasinaudojus kvadratinių uždavinių sekos rezultatais. Dėl ribojimų iškilumo šie yra aproksimuojami pritaikius išorinę aproksimaciją. Šie metodai yra realizuoti C++ su CPLEX 12.1, pritaikytu kvadratinėms aproksimacijoms surasti. Testiniai uždaviniai buvo generuojami atsitiktinai, autorių sukurtu C++ kodu. Atlikti skaitiniai eksperimentai palyginti su žinomais testais.