A node formulation for multistage stochastic programs with endogenous uncertainty

Pantuso, Giovanni

Published in:
Computational Management Science

DOI:
10.1007/s10287-021-00390-z

Publication date:
2021

Document version
Early version, also known as pre-print

Document license:
Other

Citation for published version (APA):
Pantuso, G. (2021). A node formulation for multistage stochastic programs with endogenous uncertainty. Computational Management Science, 18(3), 325 - 354. https://doi.org/10.1007/s10287-021-00390-z
A Node Formulation for Multistage Stochastic Programs with Endogenous Uncertainty

Giovanni Pantuso
Department of Mathematical Sciences
University of Copenhagen, 2100 Copenhagen, Denmark
gp@math.ku.dk
February 16, 2021

Abstract
This paper introduces a node formulation for multistage stochastic programs with endogenous (i.e., decision-dependent) uncertainty. Problems with such structure arise when the choices of the decision maker determine a change in the likelihood of future random events. The node formulation avoids an explicit statement of non-anticipativity constraints, and as such keeps the dimension of the model sizeable. An exact solution algorithm for a special case is introduced and tested on a case study. Results show that the algorithm outperforms a commercial solver as the size of the instances increases.

1 Introduction
Multistage stochastic programs offer a viable framework for modeling and solving problems involving a sequence of decisions interspersed with partial resolutions of some stochastic process. At each decision stage the decision maker knows the content of the uncertainty resolved until that stage and a probabilistic characterization of the remaining stochastic process. In the classical settings, it is assumed that decisions do not modify the stochastic process in any way (see e.g., Kall and Wallace 1994; Birge and Louveaux 1997). In other words, the uncertainty is entirely exogenous. While this description fits a large number of decision problems, several other can be found where decisions have an influence on the remainder stochastic process, e.g., by changing the likelihood of future realizations. This category of problem is referred to as multistage stochastic programs with endogenous uncertainty (MSPEU).

Following (Goel and Grossmann, 2006), there exist at least two ways in which decisions can influence the underlying stochastic process. The first possibility is that decisions alter the probability distribution of the stochastic process, thus changing the likelihood of the possible events. The second possibility is that decisions determine the time when the uncertainty is (partially) resolved. This article is concerned with multistage stochastic programs affected by the first type of endogenous uncertainty.

The research dealing with decisions influencing probability distributions is rather sparse. Jonsbråten et al. (1998) consider a case where decisions influence both the probability measure and the timing of the observation, i.e., at which stage certain random variables will be observed. The framework includes both two-stage and multistage problems, though the decisions influencing the uncertainty must be made at the first-stage only. Ahmed (2000) illustrates examples of problems with endogenous uncertainty, such as facility location, network design and server selection. The author presents an exact solution method for the resulting one-stage integer problems. Viswanath et al. (2004) consider the problem of investing in strengthening actions for the links of a network subject to disruptive events. The problem is modeled as a two-stage stochastic program where first-stage investment decisions influence the likelihood of disruptive events happening at upgraded links. The same problem is studied also by da Costa Flach (2010), Peeta et al. (2010) and Laumanns et al. (2014). Held and Woodruff (2005) consider the problem of interdicting a stochastic network, that is a network whose structure is unknown to the interdictor. In this problem, the probabilities of different future network configurations depend on previous interdiction actions. Tong et al. (2012) present an oil refinery planning problem considering that the uncertainty in product yield is...
influenced by operation mode changes. Hellemo (2016) and Hellemo et al. (2018) discuss several ways of incorporating the influence of decision variables on the underlying probability distributions in two-stage stochastic programs. Particularly, the authors formulate two-stage models where prior probabilities are distorted through an affine transformation, or combined using a convex combination of several probability distributions. Furthermore, the authors present models which incorporate the parameters of the probability distribution as first-stage decision variables. Finally, Escudero et al. (2018) study the problem of mitigating the effects of natural disasters through preventive actions. The problem is formulated as a three-stage stochastic bilinear integer program with both exogenous and endogenous uncertainty. Particularly, decisions can influence both the probabilities and the intensity of future uncertain events.

When the second type of uncertainty is considered, the decision making process is such that the uncertainty is not resolved automatically at each decision stage as in classical stochastic programs with only exogenous uncertainty. Rather, the decisions made implicitly determine the time when the uncertainty is resolved. A typical example, provided by Goel and Grossmann (2004) and based on the petrochemical industry is as follows. A decision maker has to decide which gas reservoirs to explore, and when, by installing exploration facilities. The size and quality of the reservoirs is uncertain and can be known only when facilities have been installed. Thus, the time the uncertainty is resolved depends on installation decisions.

The literature dealing with this type of stochastic programs includes Colvin and Maravelias (2008), Tarhan and Grossmann (2008), Tarhan et al. (2009), Colvin and Maravelias (2010), Gupta and Grossmann (2011), Mercier and Van Hentenryck (2011), Tarhan et al. (2013), Apap and Grossmann (2016).

The contribution of this paper is as follows. First, we introduce a novel node-formulation for multistage stochastic programs where decisions (at all stages) influence probability distributions at future stages. A node formulation automatically ensures non-anticipative decisions and thus avoids writing explicit non-anticipativity constraints (NACs) that are necessary when the uncertainty is represented via scenarios. NACs have typically been addressed as a bottleneck of available models for MSPEU, and motivated recent research to find and discard redundant NACs, see Apap and Grossmann (2016), Hooshmand and MirHassani (2016), and Hooshmand and MirHassani (2018). The new formulation is based on a novel scenario tree structure which incorporates the possibility that several (finitely many) different distributions for later stages can emanate based on the decisions made at a certain decision stage. The new scenario tree structure represents the second contribution of this paper. Third, we propose a general-purpose efficient algorithm for a special class of MSPEUs. We demonstrate the use of the new formulation, scenario tree structure and solution algorithm on instances of the Football Team Composition Problem. The instances are made available online for the benefit of future research at https://github.com/GioPan/instancesFTCPwithEndogenousUncertainty.

The remainder of this paper is organized as follows. In Section 2, we provide a model formulation and new scenario tree structure for MSPEUs. In Section 3, we describe a solution algorithm for a special case of MSPEUs. In Section 4, we present a case study in which we formulate the Football Team Composition Problem as a MSPEU and solve it using our specialized algorithm. Finally, we provide concluding remarks in Section 5.

2 Multistage stochastic programs with decision-dependent uncertainty

The decision maker is concerned with a sequence of decisions \((x_t)_{t=1}^T\) at decision stages \(t = 1, \ldots, T\), conditional on the partial resolution of a random process \((\xi_t)_{t=1}^T\). At decision stage \(t\), decisions are non-anticipative, meaning that they are based only on the realization of the random process up to, and including, \(\xi_t\). The realization of the remaining random process \(\xi_{t+1}, \ldots, \xi_T\) is still uncertain. For \(t = 2, \ldots, T\), the probability distribution of \(\xi_t, \ldots, \xi_T\) is dependent on past decisions \(x_1, \ldots, x_{t-1}\). The resulting multistage stochastic program is thus characterized by endogenously defined uncertainty.

Consistently with Jonsbraten et al. (1998), we assume that the set of potential probability distributions enforced by decisions is finite and countable. Furthermore, we assume that the probability distributions are discrete (possibly after a scenario generation phase). The latter assumption is rather standard and required in order to solve real-life stochastic programs (except perhaps for a number of specific applications). The resulting decision-dependent discrete stochastic process can be depicted by means of the scenario tree structure illustrated in Figure 1. In what follows we refer to this scenario tree structure as a multi-distribution
scenario tree (MDST) (see, e.g., also how Kaut et al. (2014) modified the classical structure of scenario trees in order to account for multiple time resolutions).

In an MDST the root node, arbitrarily named 0, represents the current state of the world, when first-stage decisions are made. First-stage decisions will enforce one out of a finite number of distributions represented by the set $D_0$. As an example, in Figure 1 decisions might determine, among other, distributions $d \in D_0$ or $d + 1 \in D_0$. Distribution $d$ is characterized by realizations, represented by nodes, that include $l$ and $m$, while distribution $d + 1$ is characterized by realizations that include $q$ and $o$. Similarly, at stage $t = 2$, assuming realization $m$ of distribution $d$ occurs at stage $t = 1$, the actions of the decision maker will determine one out of a number of different distributions including $d$ and $d + 1$ from the set $D_m$. This process continues in a similar manner until stage $T - 1$. Given a realization $n$ at stage $T - 1$ the actions of the decision maker will determine one out of a finite number of distributions $D_n$, which include $d$ and $d + 1$. Finally, at stage $T$ all the uncertainty is resolved and the decision maker makes final decisions.

Given an MDST, let us introduce the notation necessary for formulating the MSPEU. Let $\mathcal{N}$ be the set of nodes in the scenario tree, $\mathcal{N}_t$ be the set of nodes at stage $t$, $0$ the root node, $t(n) \in \{1, \ldots, T\}$ the stage of node $n$, and $a(n) \in \mathcal{N}$ the parent node of node $n$ except the root node. Let $D_n$ be the set of possible
distributions which can be enforced by decision made at node \( n \in \mathcal{N} \), and \( \mathcal{N}_{nd} \) be the child nodes of node \( n \) if distribution \( d \in \mathcal{D}_n \) is enforced. Let \( \pi_n \) be the probability of node \( n \) with \( \pi_0 = 1 \), and \( \sum_{m \in \mathcal{N}_{dn}} \pi_m = \pi_n \) for all \( n \in \mathcal{N} \), \( d \in \mathcal{D}_n \). An example of this notation is provided in Figure 2 for a three-stage scenario tree, with two possible distributions emanating from each node and each distribution being characterized by two possible realizations.

Let decision variables \( x_n \in \mathbb{R}^{\mathcal{N}_n(n)} \), \( n \in \mathcal{N} \) represent decisions made at node \( n \). These decisions may be integer or fractional and represent ordinary decision made in the decision process. Let \( \delta_{nd} \) be a binary variable which captures the probability distribution enforced by the decisions made at node \( n \). It takes value 1 if the decisions at node \( n \) enforce probability distribution \( d \) at the child nodes of \( n \), 0 otherwise. Finally, let \( \theta_n \in \mathbb{R}^1 \) be a decision variable which holds the expected value of the decisions made at the nodes descending from \( n \). For the reader’s convenience, the notation is also reported in Appendix A in a tabular format. An MSPEU is thus

\[
\begin{align*}
\max \; & r^T_0 x_0 + \sum_{d \in \mathcal{D}_0} q_{0d} \delta_{0d} + \theta_0 \\
\text{s.t.} \; & \sum_{d \in \mathcal{D}_n} \delta_{nd} = 1 \\
& n \in \mathcal{N},
\end{align*}
\]

Figure 2: Example multidistribution scenario tree with three stages, two possible distributions for each node, and two possible realizations for each distribution.
where, for each \( n \in \mathcal{N}, r_n \in \mathbb{R}^{N_t(n)} \) and \( q_{nd} \in \mathbb{R}^1 \) represent the rewards of decisions \( x_n \) and \( \delta_{nd} \), respectively; \( A_n \in \mathbb{R}^{M_t(n) \times N_t(n)} \), \( B_{nd} \in \mathbb{R}^{M_t(n) \times 1} \), \( C_{a(n)} \in \mathbb{R}^{M_t(n) \times N_t(a(n))} \), and \( D_{a(n), d} \in \mathbb{R}^{M_t(n) \times 1} \) are matrices of coefficients and \( h_n \in \mathbb{R}^{M_t(n)} \) a right-hand-side vector, with the assumption that \( C_{a(0), d} := 0 \) and \( D_{a(0), d} := 0 \). Finally, \( \Theta_n \in \mathbb{R}^1 \) represents the future expected value at leaf node \( n \), and \( X_{t(n)} \subseteq \mathbb{R}^{N_t(n)} \) the domain of the \( x_n \) variables. Objective function \((1a)\) represents the sum of the profit for the decisions made at the root node \((n = 0)\) and expected profit of future decisions. Constraints \((1b)\) ensure that the decisions made at each node determine exactly one probability distribution. That is, the stochastic phenomena following the decisions at node \( n \), are described by exactly one probability distribution (among the available ones), enforced by the decisions made. Constraints \((1c)\) describe the dependency between decision variables \( x \) and \( \delta \) at each node, and between them and the corresponding decision variables at the parent node. That is, the choice of a probability distribution at node \( n \), \( \delta_{nd} \), depends on the decisions \( x_n \) made at the same node as well as on the decisions \( x_{a(n)} \) made at the parent node and on the consequent choice of a probability distribution, \( \delta_{a(n), d} \). Constraints \((1d)\) ensure that decision variables \( \theta_n \) hold the expected value of future decisions calculated according to the probability chosen. Consider a generic node \( n \) other than a leaf node, and note that, according to \((1d)\); at this node there will be exactly one \( \delta_{nd} \) equal to one, that is exactly one distribution be chosen. Thus, the right-hand-side of constraints \((1d)\) will be equal to the term of the outer summation corresponding to the index \( d \) whose \( \delta_{nd} \) is set to one. The remaining terms are zero. Correspondingly, \( \theta_n \) will hold the expected value calculated according to the chosen probability distribution \( d \). Constraints \((1e)\) can be linearized using standard techniques. A linear reformulation is provided in Appendix B and a general procedure to determine the necessary big-M values is described in Appendix C. Constraints \((1f)\) set the future expectation at the leaf nodes. Finally, constraints \((1g)\) to \((1h)\) set the domain of the decision variables. Particularly, \( X_{t(n)} \) represents the domain of the \( x_n \) variables and may impose integrality restrictions on some/all variables.

A node formulation implicitly includes non-anticipativity, that is, automatically ensures that the decisions made at a given stage are only based on available information. On the other hand, a classical scenario formulation requires non-anticipativity explicitly enforced by means of constraints, and in general, generates a much larger problem. This can be illustrated by the following example. Consider Figure 3 which provides the scenario representation of the example MDST in Figure 2. The number of scenarios is 15, and is the same as the number of leaf nodes in Figure 2. Assuming the decision at each stage are represented by \( N \) decision variables and have to satisfy \( M \) constraints, the corresponding scenario formulation would include

- \( N \times 15 \times 3 \) decision variables, where 3 is the number of stages,
- \( M \times 15 \times 3 \) constraints, and
- approximately \( N \times (15 + 8) \) non-anticipativity constraints (approximately 15 for the first stage and 8 for the second stage, though more efficient specifications may be possible).

The node formulation would include

- \( N \times 21 \) variables (where 20 is the number of nodes in the scenario tree in Figure 2), and
- \( M \times 21 \) constraints.

Clearly, node formulations generate, in general, much smaller problems with non-anticipativity constraints playing an important role in scaling up the dimension of a scenario formulation, see e.g., [Apap and Grossmann (2016), Hooshmand and MirHassani (2016) and Hooshmand and MirHassani (2018)] for how to reduce the number of non-anticipativity constraints.
Figure 3: Scenarios in the example multidistribution scenario tree in Figure 2. Plain lines connect the nodes belonging to the same scenario. Dashed lines represent non-anticipativity constraints.
3 A solution algorithm for a special case

In this section we introduce an algorithm for solving the special case of problem (1) with \( C_{a(n)} = 0 \) for all \( n \). The algorithm builds on the fact that, when \( C_{a(n)} = 0 \), the link between decisions at subsequent stages is created by the \( \delta \) variables, i.e., those that enforce a probability distribution for the next stage. In this case, since we have finitely many distributions, we are allowed to enumerate the expected values obtainable at each node in the scenario tree. This procedure would become impractical when \( C_{a(n)} \) is different from 0, unless further assumptions on \( x_n \) are made.

Let us thus consider the following equivalent formulation of problem (1).

\[
z = \max \ x^T_0 x_0 + \sum_{d \in \mathcal{D}_0} q_{0d} \delta_{0d} + \theta_0 \tag{2a}
\]

\[
\text{s.t. } \sum_{d \in \mathcal{D}_n} \delta_{nd} = 1 \quad n \in \mathcal{N}, \tag{2b}
\]

\[
A_n x_n + \sum_{d \in \mathcal{D}_m} B_{nd} \delta_{nd} + \sum_{d \in \mathcal{D}_{a(n)}} D_{a(n), d} \delta_{a(n), d} = h_n \quad n \in \mathcal{N}, \tag{2c}
\]

\[
\theta_n \leq \phi_{nd} + M_{nd}(1 - \delta_{nd}) \quad n \in \mathcal{N} \setminus \mathcal{N}_T, d \in \mathcal{D}_n \tag{2d}
\]

\[
\phi_{nd} = \sum_{m \in \mathcal{N}_{nd}} \pi_m (x^T_m x_m + \sum_{d' \in \mathcal{D}_m} q_{md'} \delta_{md'} + \theta_m) \quad n \in \mathcal{N} \setminus \mathcal{N}_T, d \in \mathcal{D}_n \tag{2e}
\]

\[
\theta_n = \Theta_n \quad n \in \mathcal{N}_T, \tag{2f}
\]

\[
x_n \in X_{t(n)} \quad n \in \mathcal{N}, \tag{2g}
\]

\[
\delta_{nd} \in \{0, 1\} \quad n \in \mathcal{N}, d \in \mathcal{D}_n, \tag{2h}
\]

\[
\theta_n \in \mathbb{R} \quad n \in \mathcal{N}, \tag{2i}
\]

\[
\phi_{nd} \in \mathbb{R} \quad n \in \mathcal{N} \setminus \mathcal{N}_T, d \in \mathcal{D}_n. \tag{2j}
\]

Problem (2) modifies problem (1) in two elements. First, constraints (2c) take into account that \( C_{a(n)} = 0 \). Second, constraints (2d) have been linearized using constants \( M_{nd} \) and auxiliary decision variables \( \phi_{nd} \) yielding constraints (2d) and (2e) (see e.g., Appendix C for a general purpose procedure to determine these big-M values).

Problem (2) can be solved in a dynamic programming fashion using the following backward procedure. The procedure starts by calculating the optimal last-stage expectation for each node at the second-last stage and for each possible probability distribution. For each \( m \in \mathcal{N}_{T-1} \) and probability distribution \( k \in \mathcal{D}_m \), the optimal expectation at the last stage is obtained by solving the following problem.

\[
\Phi_{mk} = \max \sum_{n \in \mathcal{N}_{mk}} \pi_n (x^T_n x_n + \sum_{d \in \mathcal{D}_n} q_{nd} \delta_{nd} + \Theta_n) \tag{3a}
\]

\[
\text{s.t. } \sum_{d \in \mathcal{D}_n} \delta_{nd} = 1, \quad n \in \mathcal{N}_{mk}, \tag{3b}
\]

\[
A_n x_n + \sum_{d \in \mathcal{D}_m} B_{nd} \delta_{nd} = h_n - D_{mk} \quad n \in \mathcal{N}_{mk}, \tag{3c}
\]

\[
x_n \in X_T \quad n \in \mathcal{N}_{mk}, \tag{3d}
\]

\[
\delta_{nd} \in \{0, 1\} \quad n \in \mathcal{N}_{mk}, d \in \mathcal{D}_n. \tag{3e}
\]

Problem (3) provides the last-stage expectation for each node \( m \in \mathcal{N}_{T-1} \) at the second-last stage, and for each possible selection of a probability distribution from \( \mathcal{D}_m \). Notice in (3) that \( \Theta_n \) is input data and that \( D_{mk} \) is moved to the right-hand-side to stress that it does not multiply a decision variable as in (2c), since the choice of a probability distribution at the parent node \( m \) has been fixed to \( k \).

Then, for stage \( t = T - 2, \ldots, 1 \), for node \( m \in \mathcal{N}_t \), and for distribution \( k \in \mathcal{D}_m \) we calculate \( \Phi_{km} \) by solving problem (4).

\[
\Phi_{mk} = \max \sum_{n \in \mathcal{N}_{mk}} \pi_n (x^T_n x_n + \sum_{d \in \mathcal{D}_n} q_{nd} \delta_{nd} + \Theta_n) \tag{4a}
\]
\[ \sum_{d \in D_n} \delta_{nd} = 1, \quad n \in \mathcal{N}_{mk}, \quad (4b) \]

\[ A_n x_n + \sum_{d \in D_n} B_n d \delta_{nd} = h_n - D_{mk} \quad n \in \mathcal{N}_{mk}, \quad (4c) \]

\[ \theta_n \leq \Phi_{nd} + M_{nd} (1 - \delta_{nd}), \quad d \in D_n, n \in \mathcal{N}_{mk}, \quad (4d) \]

\[ x_n \in X_{t(n)} \quad n \in \mathcal{N}_{mk}, \quad (4e) \]

\[ \delta_{nd} \in \{0, 1\} \quad n \in \mathcal{N}_{mk}, d \in D_n, \quad (4f) \]

\[ \theta_n \in \mathbb{R} \quad n \in \mathcal{N}_{mk}. \quad (4g) \]

Notice in problem (4) that \( \Phi_{nd} \) is input data and has been calculated in the previous steps of the algorithm.

Finally, we can solve the following problem for the root node.

\[ z = \max r_T^T x_0 + \sum_{d \in D_0} q_{0d} \delta_{0d} + \theta_0 \quad (5a) \]

\[ \sum_{d \in D_0} \delta_{0d} = 1, \quad (5b) \]

\[ A_0 x_0 + \sum_{d \in D_0} B_{0d} \delta_{0d} = h_0, \quad (5c) \]

\[ \theta_0 \leq \Phi_{0d} + M_{0d} (1 - \delta_{0d}), \quad d \in D_0, \quad (5d) \]

\[ x_0 \in X_1, \quad (5e) \]

\[ \delta_{0d} \in \{0, 1\} \quad d \in D_0, \quad (5f) \]

\[ \theta_0 \in \mathbb{R}. \quad (5g) \]

Letting \( D = \max_{n \in \mathcal{N}} |D_n| \), the algorithm entails solving \( O(D|N|) \) mixed-integer programs of size significantly smaller than (1).

4 Case study

In this section we present a case study based on the Football Team Composition Problem (FTCP – Pantuso (2017); Pantuso and Hvattum (2020)) which we extend in order to account for decision-dependent uncertainty. The problem consists of selecting players for a football team while their future market value is uncertain and influenced by the team for which they play. The scope of the computational study is to compare the performance of the algorithm to that of a state-of-the-art commercial solver. We remark, however, that formulation (1) and the solution algorithm proposed are general and applicable beyond the context of the FTCP, which we use solely as an example. Decision problems under endogenous uncertainty may, in fact, arise in several business contexts, some of which are mentioned in Section 1. To provide another practical example, consider an agribusiness making periodic production planning decisions for a number of different crops while demand is uncertain. Product substitution is a common phenomenon in the production of crops. In fact, a customer, say a farmer, may view multiple crops as suitable for their farming needs, see, e.g., Bansal and Dyer (2020). Thus, the uncertain demand for a crop may depend in part on the portfolio of crops an agribusiness offers for sale, yielding a decision problem under endogenous uncertainty.

In Section 4.1 we describe the FTCP in more details and formulate it as an MSPEU. In Section 4.2 we illustrate an efficient procedure to obtain big-\( M \) values. In Section 4.3 we introduce the problem instances and finally in Section 4.4 we present and discuss the results. The data of the problem instances is available online at [https://github.com/GioPan/instancesFTCPwithEndogenousUncertainty](https://github.com/GioPan/instancesFTCPwithEndogenousUncertainty) for the benefit of future research.

4.1 The Football Team Composition Problem

The FTCP is the problem of composing a football team by purchasing and selling professional football players. A complete description of the problem can be found in Pantuso (2017). In what follows, we report
the basic elements necessary for this case study. The decision problem can be described as follows. At every stage, i.e., transfer market window (TMW), professional football clubs can renew their teams by purchasing players from other clubs or selling available players to other clubs. In order to participate in national and international competitions, professional clubs must compose a team made of a fixed number of players. In addition, the coach of the team requires players with different roles (e.g., defenders or mid-fielders) and skills.

Club managers are often given a budget to spend in the transfer market and are typically allowed to reinvest the revenue from the sale of players. The current market value of football players is known, while the future value is stochastic as it depends on a number of random events such as injuries, fitness, motivation and ultimately luck. Therefore, at every TMW, football clubs make decisions in conditions of uncertainty with the scope of maximizing the expected value of the team. Furthermore, the market values of the players in the same team are strongly correlated. This generates a multistage stochastic program with endogenous uncertainty, since the decision of hiring or selling a player will change the correlations of the joint value distribution of the players considered. That is, hiring a football player will make their value strongly correlated with the value of the other players in the team, while selling or not hiring a player will make their value uncorrelated with the value of the players in the team.

In order to model the FTCP we assume the club is evaluating a finite number of alternative team compositions. Every team composition consists of the required number of players and contains the necessary mix of skills. Let $I$ be the set of possible team compositions, $N$ the set of nodes in the MDST describing the underlying uncertainty, $N_{in}$ the set of child nodes of node $n \in N$ if team composition $i \in I$ is chosen at node $n$. Note in fact that the selection of a team composition determines the correlations between the values of the players in the instance, and thus the probability distribution. Let $N^L$ be the set of leaf nodes, i.e., the nodes at the last decision stage, and $t(n) \in \{1, \ldots, T\}$ the stage corresponding to node $n$.

Let $V_i$ be the value of team composition $i \in I$ at node $n$ (i.e., the sum of the values of the players in team composition $i$), let $C^S_{in}$ be the cost of the salary of the players in team composition $i$ at node $n$, $C^T_{jn}$ the cost of transitioning from team composition $i$ to team composition $j$ at node $n$, i.e., the cost of the transfer fees for the players bought minus the revenue for the transfer fees of the players sold. Furthermore, let $B_i$ be the budget available to the club for the transfer market at node $n$ and $E_n$ a stochastic extra-budget conditional on events such as sport successes. Let $X_i$ be equal to one if team composition $i \in I$ is the initial team composition, $M_i$ a suitably large constant for each $i$ and $n$ (see Section 4.2), and $\rho$ a discount rate. Let $\delta_{in}$ be a binary decision variable which is equal 1 if team composition $i \in I$ is chosen at node $n \in N$, $x_{ijn}$ a binary decision variable which is equal 1 if the club transitions from team composition $i \in I$ to team composition $j \in J$ at node $n \in N$ and, $\phi_{in}$ the expected net value of the team at the children of node $n$ if composition $i$ is chosen and, finally, $\theta_n$ the expected net value of the team at the child nodes of node $n \in N$. The FTCP with endogenous uncertainty is hence:

\[
\begin{align*}
\max \sum_{i \in I} \left( V_i \delta_{i0} - C^S_{i0} \delta_{i0} - \sum_{j \in I} C^T_{ij0} x_{ij0} \right) + \theta_0 \\
\text{s.t.} \quad \sum_{i \in I} \delta_{in} &= 1 \\
\delta_{i0} - X_i + \sum_{j \in I : j \neq i} (x_{ij0} - x_{j0}) &= 0 \\
\delta_{in} - \delta_{i,a(n)} + \sum_{j \in I : j \neq i} (x_{ijn} - x_{jin}) &= 0 \\
\sum_{i \in I} \sum_{j \neq i} x_{ijn} &\leq 1 \\
\end{align*}
\]
\[
\sum_{i \in I} \sum_{j \in I} C_{ij}^T x_{ijn} \leq B + E_n 
\]

\[ n \in \mathcal{N}, \]  

(6f)

\[ \theta_n \leq \phi_{in} + M_{in}(1 - \delta_{in}) \]

\[ i \in \mathcal{I}, n \in \mathcal{N}, \]  

(6g)

\[ \phi_{in} = \frac{1}{1 + \rho(n)} \pi \left\{ \theta_m + \sum_{j \in \mathcal{I}} \left( V_{jm} \delta_{jm} - C_{jm}^S \delta_{jm} - \sum_{k \in \mathcal{I}} C_{jkm}^T x_{jkm} \right) \right\} \]

\[ i \in \mathcal{I}, n \in \mathcal{N} \setminus \mathcal{N}^L, \]

(6h)

\[ \theta_n = \frac{1}{1 + \rho(n)+1} \sum_{i \in \mathcal{I}} V_{in} \delta_{in} \]

\[ n \in \mathcal{N}^L, \]

(6i)

\[ \delta_{in} \in \{0, 1\} \]

\[ i \in \mathcal{I}, n \in \mathcal{N}, \]

(6j)

\[ x_{ijn} \in \{0, 1\} \]

\[ i, j \in \mathcal{I}, n \in \mathcal{N}, \]

(6k)

\[ \theta_n \in \mathbb{R} \]

\[ n \in \mathcal{N}. \]  

(6l)

Objective function (6a) represents the sum of the value of the team composition chosen here-and-now, minus the expenses for salaries and, if any, the transition cost from the initial team composition. In addition it takes into account the expected value of the team at future nodes. Constraints (6b) ensure that only one team composition is chosen at each decision node, while constraints (6c)-(6d) state that either the club holds the same team composition as in the previous season or a new team composition is chosen at node 0 and at the rest of the nodes, respectively. Constraints (6e) state that at most one team change can be done at each node. Constraints (6f) state that the future net expected value of the team depends on the team composition chosen. Constraints (6g) sets the expected value at the children of node \( n \) if composition \( i \) is chosen, while constraints (6i) set the final value of \( \theta_n \) to the value of the team chosen at the leaf nodes. This corresponds to a sunset value which accounts the termination of an infinite horizon problem. Notice that the final value of the team is discounted as a future value. Finally, constraints (6j)-(6l) define the domain for the decision variables.

### 4.2 Finding big-M values for the FTCP

We illustrate a fast method to set the \( M_{in} \) values in (6). The method proposed resulted, on this special case, significantly faster than the general method described in Appendix C.

We start by observing that, for all \( i \in \mathcal{I} \) and \( n \in \mathcal{N} \) we are looking for a value \( M_{in} \) such that:

\[ \theta_n - \phi_{in} \leq M_{in} \]

and that, based on constraints (6g), we have

\[ \theta_n \leq \max_{i \in \mathcal{I}} \phi_{in} \]

consequently, we need to set

\[ M_{in} \geq \max_{i \in \mathcal{I}} \phi_{in} - \phi_{in} \]

thus, under the very mild assumption that \( \phi_{in} \geq 0 \) (i.e., that the team does not spend in transfers and salaries more than the value of the entire team), for every \( i \in \mathcal{I} \) a valid \( M_{in} \) is

\[ M_{in} = M_n = \max_{i \in \mathcal{I}} \phi_{in} \]

By expanding \( \phi_{in} \) according to constraints (6h) and (6i) we can set...
\[ M_{in} = \max_{i \in I} \frac{1}{1 + \rho(t)} V_{in} \text{ for all } i \in I \text{ if } n \in N^L \]

\[ M_{in} = \max_{i \in I} \left\{ \sum_{m \in N_{in}} \frac{1}{1 + \rho(t)} \pi_m \left( M_m + \max_{j \in I} \{ V_{jm} \} \right) \right\} \text{ for all } i \in I \text{ if } n \in N \setminus N^L. \]

### 4.3 Problem instances

Instances are generated from the case studies on the FTCP presented by Pantuso (2017) and are available online at [https://github.com/GioPan/instancesFTCPwithEndogenousUncertainty](https://github.com/GioPan/instancesFTCPwithEndogenousUncertainty). The instances consist of the 20 teams competing in the English Premier League 2013/2014 and dealing with the transfer market of summer 2014. Each team is characterized by a budget, a list of players (made of the current and target players), and a number of randomly generated team compositions \( I \) complying with the regulations of the league and with the coach’s specifics. In turn, each player is characterized by age, role, current market value, salary, selling and purchase price. The value and cost of salaries of a team composition are calculated as the sum of values and salaries, respectively, of the players contained. Similarly, transition costs are calculated as the revenue for the players sold, minus the cost of the players bought.

Future player values are stochastic and modeled by means of the regression equation available in Pantuso (2017). The joint probability distribution of the market values for all the players forms a multivariate normal distribution. In addition, we set a 0.8 correlation between the players belonging to the focal team and no correlation between the players belonging to the focal team and the remaining players. Therefore, the correlations change with the decisions of the team to buy or sell players creating an MSPEU. Scenario trees are obtained by sampling realizations from the underlying multivariate normal distributions. Finally, for each node in the scenario tree, the random extra-budget \( E_n \) is calculated as a percentage of the deterministic budget \( B \). The percentage is the same as percentage increase of team value from the parent node, that is \( \max \{0, 100 \times (V_{in} - V_{i,a(n)}) / V_{i,a(n)}\} \). That is, an increase in the team value yields an increase in the spending capability of the team.

The instances represent a four-stage horizon and are identified by (i) the focal team, where Team \( \in \{ \text{ARS, ASV, CAR, CHE, CRP, EVE, FUL, HUL, LIV, MAC, MAU, NEC, NOR, SOU, STO, SUN, SWA, TOT, WBA, WHU} \} \), (ii) the number of team composition \( |I| \in \{3, 4, 5\} \), and (iii) the number of samples in each distribution \( S \in \{4, 5\} \), yielding 120 numerically different instances with up to approximately 400 thousand binary variables.

### 4.4 Results

We implemented our algorithm in Python 2.7 using Cplex 12.8 for solving the subproblems. Cplex 12.8 has also been used to solve the full problems. All experiments have been run on a server with 64 double AMD Opteron 6380 processors and 251 GB RAM. Tables 1 to 3 report the results for different cardinalities \( |I| \) and different number of samples \( S \) for all the focal teams. The computation times do not include the time required to find big-M values as illustrated in Section 4.2. An account of these times is reported in Appendix D. Note that big-M values are required both when solving the full problem and when using our algorithm and in both cases are calculated beforehand. Thus, their computation time does not have an impact on the comparison between the two solution methods. Since the algorithm entails iteratively building and solving mixed-integer programs, the times reported include the time required for building the models.

Table 1 reports the results with \( |I| = 3 \), generating instances with up to approximately 32000 binary variables. For these instances it can be noticed that Cplex performs much better than our method on the smaller test cases (with \( S = 4 \) samples) while our algorithm is more competitive on the instances with \( S = 5 \). Altogether, Cplex solves the problems approximately 25% faster than our method.
By increasing the number of team compositions we generate larger MDSTs, and the corresponding MSPEUs also become larger. Table 2 reports the results with $|I| = 4$ generating a MDST with four possible distributions at each decision node and problems with up to approximately 135 thousand binary variables. In this case, our algorithm performs significantly better that Cplex on almost all instances, solving the
This pattern illustrates that our algorithm scales significantly better than Cplex, and is confirmed also by our results on the instances with $|Z| = 5$, the largest we tested. These instances generate problems with up
to approximately 400 thousand binary variables and the corresponding results are provided in Table 3. It is possible to notice that our algorithm outperforms Cplex on almost all instances. Altogether, our algorithm is able to solve the same instances 61.1% faster than Cplex. Only on two instances (CAR and CHE, with \( S = 4 \)) Cplex performs better than our algorithm. Similar cases are to be expected since our algorithm entails solving several subproblems which are mixed-integer programs. Therefore, it is possible that some numerically difficult subproblems are found that slow down the entire algorithm (in our case we do not solve problems in parallel). However, despite that, the average performance of our algorithm on the largest instance is by far better.

These results illustrate that MSPEU easily generate very large optimization problems. The size of the problems, in our instances, increases by approximately three to five times by increasing the number of possible distributions. However, the algorithm we provide scales better than the solver Cplex. Particularly, when using Cplex, the average solution time increases by approximately 330% when increasing \(|I|\) from 3 to 4, and by approximately 175% when increasing \(|I|\) from 4 to 5. With our algorithm the average solution time increases by approximately 203% when increasing \(|I|\) from 3 to 4, and by approximately 158% when increasing \(|I|\) from 4 to 5.

Table 3: Results with \(|I| = 5\). \( S \) indicates the number of realizations describing each distribution. \#Var, \#Bin and \#Con indicate the total number of variables, the number of binary variables, and the number of constraints, respectively. \( \tau \) (Cpx) indicate the elapsed time when using the algorithm (Cplex). Obj. (Cpx) indicates the objective value obtained using the algorithm (Cplex). \( \Delta \tau \) is calculated as \( 100(\tau - \tau \text{ Cpx})/\tau \text{ Cpx} \).

| Case | \( S \) | \#Var | \#Bin | \#Con | \( \tau \) Cpx [sec] | Obj. Cpx | \( \tau \) [sec] | Obj. | \( \Delta \tau \) [%] |
|------|------|------|------|------|------------------|---------|---------------|-----|------------|
| ARS  | 4    | 261051 | 210525 | 151578 | 4148.394         | 1412.322 | 2487.630      | 1412.322 | -40.034    |
| ASV  | 4    | 261051 | 210525 | 151578 | 9128.747         | 888.394  | 2360.369      | 888.394  | -74.404    |
| CAR  | 4    | 261051 | 210525 | 151578 | 2324.632         | 559.909  | 2951.872      | 559.909  | 26.982     |
| CHE  | 4    | 261051 | 210525 | 151578 | 2941.786         | 2351.709 | 4074.808      | 2351.709 | 38.515     |
| CRP  | 4    | 261051 | 210525 | 151578 | 4873.336         | 403.008  | 2757.630      | 403.008  | -43.408    |
| EVE  | 4    | 261051 | 210525 | 151578 | 9240.398         | 1318.359 | 2367.225      | 1328.380 | -74.382    |
| FUL  | 4    | 261051 | 210525 | 151578 | 9385.279         | 676.308  | 2775.872      | 676.308  | -70.425    |
| HUL  | 4    | 261051 | 210525 | 151578 | 1748.831         | 1589.133 | 1070.475      | 1589.133 | -38.789    |
| MAU  | 4    | 261051 | 210525 | 151578 | 9294.710         | 2529.355 | 1935.253      | 2533.948 | -79.179    |
| NEC  | 4    | 261051 | 210525 | 151578 | 6984.354         | 1202.156 | 1502.823      | 1202.156 | -78.483    |
| NIO  | 4    | 261051 | 210525 | 151578 | 3516.805         | 420.312  | 1038.191      | 420.312  | -70.479    |
| NOR  | 4    | 261051 | 210525 | 151578 | 3593.291         | 599.703  | 947.766       | 599.703  | -73.624    |
| SOU  | 4    | 261051 | 210525 | 151578 | 12808.767        | 1473.064 | 2854.601      | 1706.759 | -77.715    |
| MAC  | 4    | 261051 | 210525 | 151578 | 5341.258         | 1681.019 | 2608.034      | 1681.019 | -80.476    |
| FUL  | 5    | 504556 | 406900 | 292968 | 5985.321         | 1453.229 | 3231.294      | 1453.229 | -61.605    |
| ARS  | 5    | 504556 | 406900 | 292968 | 13100.017        | 969.935  | 2557.630      | 969.935  | -80.476    |
| CAR  | 5    | 504556 | 406900 | 292968 | 12990.803        | 612.171  | 3115.356      | 612.171  | -76.019    |
| CHE  | 5    | 504556 | 406900 | 292968 | 15815.020        | 2608.034 | 4221.096      | 2608.034 | -73.310    |
| CRP  | 5    | 504556 | 406900 | 292968 | 13160.004        | 476.955  | 3668.813      | 476.955  | -72.121    |
| EVE  | 5    | 504556 | 406900 | 292968 | 18368.885        | 1299.376 | 2193.376      | 1299.376 | -84.140    |
| FUL  | 5    | 504556 | 406900 | 292968 | 13385.821        | 746.554  | 2913.376      | 755.759  | -78.993    |
| HUL  | 5    | 504556 | 406900 | 292968 | 12995.828        | 516.157  | 3335.790      | 516.157  | -74.332    |
| LIV  | 5    | 504556 | 406900 | 292968 | 12808.767        | 1473.064 | 2854.447      | 1706.759 | -77.715    |
| MAC  | 5    | 504556 | 406900 | 292968 | 5341.258         | 1681.019 | 2608.034      | 1681.019 | -61.605    |
| MAU  | 5    | 504556 | 406900 | 292968 | 13282.279        | 2690.920 | 3722.067      | 2712.864 | -71.977    |

14
5 Conclusions

This paper introduced 1) a novel scenario tree structure and 2) a node formulation for multistage stochastic programs with endogenous uncertainty, as well as 3) a solution algorithm for a special case. A computational study shows that while, as expected, problems with endogenous uncertainty tend to generate large optimization problems, all our instances where solvable by Cplex to optimality in at most approximately 5 hours. Furthermore, our algorithm outperformed Cplex on the medium and large instances and showed that it scales well with the size of the problem.

Despite the encouraging results obtained in our study, solving multistage stochastic programs with endogenous uncertainty remains, in general, a challenging task. Our algorithm requires an explicit scenario tree structure, and solves a number of problems which grows linearly with the number of nodes. However, the number of nodes in a scenario tree grows exponentially with the number of stages and the treatment of cases with more than a handful of stages may soon become prohibitive. New approaches in the spirit of Pereira and Pinto (1991); Zou et al. (2019), based on progressive approximations of future stages, may be proven more scalable. Furthermore, our models employ so called big-M constants. Poorly chosen big-M values, e.g., by trial-and-error, may become problematic. It is well known that they may create numerical difficulties when solving mixed-integer programs. In addition, as discussed in Pineda and Morales (2019), they may lead to highly sub-optimal solutions. The authors use a simple bilevel programming problem (which can be reformulated as a mixed-integer program that uses big-Ms) to show how a poorly designed trial-and-error procedure may generate the false belief that the solution to the reformulation is indeed optimal for the original bilevel problem. Based on these evidences we also advocate caution and the use of more sophisticated procedures for setting big-M values. The procedure in Appendix C goes in this direction. Finally, cases more general than the special one treated Section 3 remain to be addressed.

References

S. Ahmed. Strategic planning under uncertainty: Stochastic integer programming approaches. Phd, University of Illinois at Urbana-Champaign, 2000.

R. M. Apap and I. E. Grossmann. Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. Computers & Chemical Engineering, 103:233–274, 2016. ISSN 0098-1354.

S. Bansal and J. S. Dyer. Planning for end-user substitution in agribusiness. Operations Research, 68(4):1000–1019, 2020.

J. R. Birge and F. Louveaux. Introduction to stochastic programming. Springer, New York, 1997.

M. Colvin and C. T. Maravelias. A stochastic programming approach for clinical trial planning in new drug development. Computers & Chemical Engineering, 32(11):2626–2642, 2008.

M. Colvin and C. T. Maravelias. Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. European Journal of Operational Research, 203(1):205–215, 2010.
B. da Costa Flach. *Stochastic Programming with Endogenous Uncertainty: An Application in Humanitarian Logistics.* PhD thesis, PUC-Rio, 2010.

L. F. Escudero, M. A. Garín, J. F. Monge, and A. Unzueta. On preparedness resource allocation planning for natural disaster relief under endogenous uncertainty with time-consistent risk-averse management. *Computers & Operations Research*, 98:84–102, 2018.

V. Goel and I. E. Grossmann. A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. *Computers & Chemical Engineering*, 28:1409–1429, 2004.

V. Goel and I. E. Grossmann. A Class of stochastic programs with decision dependent uncertainty. *Mathematical Programming*, 108:355–394, 2006.

V. Gupta and I. E. Grossmann. A Class of stochastic programs with decision dependent uncertainty. *Mathematical Programming*, 35:2235–2247, 2011.

H. Held and D. L. Woodruff. Heuristics for Multi-Stage Interdiction of Stochastic Networks. *Journal of Heuristics*, 11(5-6):483–500, 2005.

L. Hellemo. *Managing Uncertainty in Design and Operation of Natural Gas Infrastructure.* Phd, Norwegian University of Science and Technology, 2016.

L. Hellemo, P. I. Barton, and A. Tomasgard. Decision-dependent probabilities in stochastic programs with recourse. *Computational Management Science*, 15(3):369–395, 2018.

F. Hooshmand and S. A. MirHassani. Efficient constraint reduction in multistage stochastic programming problems with endogenous uncertainty. *Optimization Methods and Software*, 31:359–376, 2016.

F. Hooshmand and S. A. MirHassani. Reduction of nonanticipativity constraints in multistage stochastic programming problems with endogenous and exogenous uncertainty. *Mathematical Methods of Operations Research*, 87(1):1–18, 2018.

T. W. Jonsbråten, R. J.-B. Wets, and D. L. Woodruff. A class of stochastic programs with decision dependent random elements. *Annals of Operations Research*, 82:83–106, 1998.

P. Kall and S. W. Wallace. *Stochastic programming.* John Wiley and Sons Ltd, Chichester, 1994.

M. Kaut, K. T. Midthun, A. S. Werner, A. Tomasgard, L. Hellemo, and M. Fodstad. Multi-horizon stochastic programming. *Computational Management Science*, 11(1-2):179–193, 2014.

M. Laumanns, S. Prestwich, and B. Kawas. Distribution shaping and scenario bundling for stochastic programs with endogenous uncertainty. In J. L. Higle, W. Römisch, and S. Sen, editors, *Stochastic Programming E-Print Series*, Stochastic Programming E-Print Series. Institut für Mathematik, 2014.

L. Mercier and P. Van Hentenryck. An anytime multistep anticipatory algorithm for online stochastic combinatorial optimization. *Annals of Operations Research*, 184(1):233–271, 2011.

G. Pantuso. The Football Team Composition Problem: A Stochastic Programming approach. *Journal of Quantitative Analysis in Sports*, 13(3):113–129, 2017.

G. Pantuso and L. M. Hvattum. Maximizing performance with an eye on the finances: a chance-constrained model for football transfer market decisions. *TOP*, pages 1–29, 2020.

S. Peeta, F. S. Salman, D. Gunnece, and K. Viswanath. Pre-disaster investment decisions for strengthening a highway network. *Computers & Operations Research*, 37:1708–1719, 2010.

M. V. Pereira and L. M. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1-3):359–375, 1991.

S. Pineda and J. M. Morales. Solving linear bilevel problems using big-ms: Not all that glitters is gold. *IEEE Transactions on Power Systems*, 34(3):2469–2471, 2019. doi: 10.1109/TPWRS.2019.2892607.
B. Tarhan and I. E. Grossmann. A multistage stochastic programming approach with strategies for uncertainty reduction in the synthesis of process networks with uncertain yields. *Computers & Chemical Engineering*, 32:766–788, 2008.

B. Tarhan, I. E. Grossmann, and V. Goel. Stochastic Programming Approach for the Planning of Offshore Oil or Gas Field Infrastructure under Decision-Dependent Uncertainty. *Industrial & Engineering Chemistry Research*, 48:3078–3097, 2009.

B. Tarhan, I. E. Grossmann, and V. Goel. Computational strategies for non-convex multistage MINLP models with decision-dependent uncertainty and gradual uncertainty resolution. *Annals of Operations Research*, 203(1):141–166, 2013.

K. Tong, Y. Feng, and G. Rong. Planning under Demand and Yield Uncertainties in an Oil Supply Chain. *Industrial & Engineering Chemistry Research*, 51(2):814–834, 2012.

K. Viswanath, P. Srinivas, and S. F. Sibel. Investing in the Links of a Stochastic Network to Minimize Expected Shortest Path Length. 2004. URL https://www.krannert.purdue.edu/programs/phd/working-papers-series/2004/1167.pdf

J. Zou, S. Ahmed, and X. A. Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175(1-2):461–502, 2019.

### A Notation table

| Sets | Parameters | Variables |
|------|------------|-----------|
| \{1, \ldots, T\} | Stage of node n | Decisions made at node n |
| \mathcal{N} | Parent node of node n | Decision on whether to apply probability distribution d at node n |
| \mathcal{N}_t \subseteq \mathcal{N} | Probability of node n | Expected value of the decisions made at the nodes descending from n |
| \mathcal{D}_n | Coefficients of decision variables \(x_n\) in the objective function | |
| \mathcal{N}_{nd} \subseteq \mathcal{N} if distribution d \in \mathcal{D}_n is enforced | Coefficients of decision variable \(\delta_{nd}\) in the objective function | |
| \mathcal{X}_t | Coefficients of variables \(x_n\) in the constraints that connect \(x_n\) and \(\delta_{nd}\) decisions | |
| \mathcal{N}_{nd} \subseteq \mathcal{N} if distribution d \in \mathcal{D}_n is enforced | Coefficients of variable \(\delta_{nd}\) in the constraints that connect \(x_n\) and \(\delta_{nd}\) decisions | |
| \mathcal{X}_t | Coefficients of variables \(x_{a(n)}\) in the constraints that connect \(x_n\) and \(\delta_{nd}\) decisions | |
| \mathcal{A}_n \in \mathbb{R}^{\mathcal{M}_l(n) \times \mathcal{N}_t(n)} | Coefficients of variable \(\delta_{nd}\) in the constraints that connect \(x_n\) and \(\delta_{nd}\) decisions | |
| \mathcal{B}_{nd} \in \mathbb{R}^{\mathcal{M}_l(n) \times \mathcal{X}_t} | Coefficients of variable \(\delta_{nd}\) in the constraints that connect \(x_n\) and \(\delta_{nd}\) decisions | |
| \mathcal{C}_{a(n)} \in \mathbb{R}^{\mathcal{M}_l(n) \times \mathcal{N}_t(a(n))} | Right-hand-side coefficients of the constraints that connect \(x_n\) and \(\delta_{nd}\) decisions | |
| \mathcal{D}_{a(n),d} \in \mathbb{R}^{\mathcal{M}_l(n) \times \mathcal{X}_t} | Terminal value of the decisions following node \(n \in \mathcal{N}_T\) | |
| \mathcal{h}_n \in \mathbb{R}^{\mathcal{M}_l(n)} | | |
| \Theta_n \in \mathbb{R} | | |

17
B  A big-$M$ reformulation

In this appendix a big-$M$ reformulation that linearizes model (1) is introduced. In addition to the notation introduced in Section 2 let $M_{nd} \in \mathbb{R}^+$ be a suitably high constant. The linearized EUMSP is thus

$$\max \ r_0^T x_0 + \sum_{d \in \mathcal{D}_n} q_{nd} \delta_{nd} + \theta_0 \quad \text{(7a)}$$

s.t. $\sum_{d \in \mathcal{D}_n} \delta_{nd} = 1 \quad \text{n} \in \mathcal{N}, \quad \text{(7b)}$

$$A_n x_n + \sum_{d \in \mathcal{D}_n} B_{nd} \delta_{nd} + C_{u(n)} x_{u(n)} + \sum_{d \in \mathcal{D}_{u(n)}} D_{u(n),d} \delta_{u(n),d} = h_n \quad \text{n} \in \mathcal{N}, \quad \text{(7c)}$$

$$\theta_n \leq \sum_{m \in \mathcal{N}_{nd}} \pi_m (r_m^T x_m + \sum_{d \in \mathcal{D}_m} q_{md} \delta_{md} + \theta_m) + M_{nd} (1 - \delta_{nd}) \quad \text{n} \in \mathcal{N} \setminus \mathcal{N}_T, \ d \in \mathcal{D}_n \quad \text{(7d)}$$

$$\theta_n = \Theta_n \quad \text{n} \in \mathcal{N}_T, \quad \text{(7e)}$$

$$x_n \in X_{i(n)} \quad \text{n} \in \mathcal{N}, \quad \text{(7f)}$$

$$\delta_{nd} \in \{0, 1\} \quad \text{n} \in \mathcal{N}, \ d \in \mathcal{D}_n, \quad \text{(7g)}$$

$$\theta_n \in \mathcal{R} \quad \text{n} \in \mathcal{N}. \quad \text{(7h)}$$

Note, particularly, that constraints (7d) are equivalent to (1d). Consider a given node $n$, other than a leaf node. Observe that only for one distribution $d$ there will be a $\delta_{nd}$ which takes value one at $n$ (see (7b)). For the same $n$ and for the same $d$, the second term on the right-hand-side of (7d) will be zero (i.e., the big-$M$ will not be enforced), and the resulting right-hand-side will be the most binding among the $|\mathcal{D}_n|$ constraints for node $n$. Since we are maximizing, at optimality $\theta_n$ will take value of the expectation according to the distribution $d$ for which $\delta_{nd} = 1$, as it happens in model (1).

C  Finding big-$M$ values

An efficient implementation of model (7) requires tight big-$M$ values. Observe that, for $n \in \mathcal{N} \setminus \mathcal{N}_T$ and $d \in \mathcal{D}_n$, constant $M_{nd}$ must be a valid upper bound for constraints (7d), that is:

$$\theta_n - \sum_{m \in \mathcal{N}_{nd}} \pi_m (r_m^T x_m + \sum_{d \in \mathcal{D}_m} q_{md} \delta_{md} + \theta_m) \leq M_{nd}$$

Let us introduce $\phi_{nd}$ to represent the expectation at the children of node $n$ for distribution $d$, that is:

$$\phi_{nd} = \sum_{m \in \mathcal{N}_{nd}} \pi_m (r_m^T x_m + \sum_{d \in \mathcal{D}_m} q_{md} \delta_{md} + \theta_m)$$

Consider the numerical example shown in Table 5 for a given node $n$ and three possible distributions $d$. The table reports the highest and lowest values the expectation at the following stage can take for each possible distribution, and the corresponding value of $\theta_n$ should a specific distribution be chosen. When choosing $M_{n,d_1}$, notice that the maximum value $\theta_n$ can reach for other distributions is 9, and that the least value $\phi_{n,d_1}$ can reach is 5. Therefore, we can set $M_{n,d_1} = 9 - 5$. In fact, if distribution $d_2$ is chosen, $\theta_n$ will be at most 9 and $\phi_{n,d_1}$ at least 5, thus adding 4 to $\phi_{n,d_1}$ will ensure that $\theta_n$ is correctly set to 9. Similarly, we choose $M_{n,d_2} = 6$ as the highest value $\theta_n$ can take if $d_2$ is not selected is 10, while the least value of $\phi_{n,d_2}$ is 4. Finally, with a similar reasoning we can set $M_{n,d_3} = 7$.

From the example in Table 5 we understand that finding values for $M_{nd}$ amounts to finding highest values for $\phi_{nd}$ and differences $\theta_n - \phi_{nd}$. In what follows we illustrate how these values can be found for $t = T - 1$ in Section C.1 and $t = T - 2, \ldots, 1$ in Section C.2.

C.1  Big-$M$ values for stages $t = T - 1$

We start at the second-last stage, $t = T - 1$. Our task is that of finding, for each node $\tilde{n} \in \mathcal{N}_{T-1}$ and for each distribution $\tilde{d} \in \mathcal{D}_{\tilde{n}}$, a constant $M_{\tilde{nd}}$ which is slightly higher than the highest difference $\theta_{\tilde{n}} - \phi_{\tilde{n}\tilde{d}}$, where
Table 5: Numerical example for the calculation of constants $M_{nd}$

| $d$ | $\theta_n$ | $\phi_{nd}$ | Max | Min | $M_{nd}$ |
|-----|-------------|--------------|-----|-----|---------|
| $d_1$ | 10          | 10           | 5   | 9   | $9 - 5 = 4$ |
| $d_2$ | 9           | 9            | 4   | 10  | $10 - 4 = 6$ |
| $d_3$ | 8           | 8            | 3   | $10 - 3 = 7$ |

Again

$$\phi_{\bar{n}d} = \sum_{m \in \mathcal{N}_{nd}} \pi_m (r_m^T x_m + \sum_{d \in \mathcal{D}_m} q_{md} \delta_{md} + \theta_m)$$

Now, the highest difference can be found solving the following optimization problem:

$$M_{\bar{n}d}^* = \max \theta_n - \sum_{m \in \mathcal{N}_{nd}} \pi_m (r_m^T x_m + \sum_{k \in \mathcal{D}_m} q_{mk} \delta_{mk} + \Theta_m)$$

s.t. \( \sum_{d \in \mathcal{D}_n} \delta_{nd} = 1 \) \( n \in \mathcal{N} \), (8b)

$$A_n x_n + \sum_{d \in \mathcal{D}_n} B_{nd} \delta_{nd} + C_{a(n)} x_{a(n)} + \sum_{d \in \mathcal{D}_a(n)} D_{a(n),d} \delta_{a(n),d} = h_n$$ \( n \in \mathcal{N} \), (8c)

$$\theta_n \leq \Theta_{\bar{n}d}^*$$ \( n \in \mathcal{N} \), (8d)

$$x_n \in X_{t(n)}$$ \( n \in \mathcal{N} \), (8e)

$$\delta_{nd} \in \{0, 1\}$$ \( n \in \mathcal{N}, d \in \mathcal{D}_n \), (8f)

$$\delta_{\bar{n}d} = 1$$ (8g)

Problem (8) consists of finding the feasible solution to problem (7) which yields the highest value for the left-hand-side of constraint (7d) for $\bar{n}$ and $\bar{d}$. The following two elements must be noted in (8). Constraints (7d) of the original problem, which determine the correct expectations at the stages before $T - 1$, are not included as they are irrelevant for stage $T - 1$. The second element to note is constraint (8d) which sets an upper bound $\theta_n$. This upper bound represents the highest value $\theta_n$ can take for the distributions other than $\bar{d}$. This value can, in turn, be obtained solving optimization problems. The highest expectation for stage $T$, given distribution $d' \in D_{\bar{n}}$ is the optimal value to problem (9):

$$\Phi_{\bar{n}d'}^* = \max \sum_{m \in \mathcal{N}_{nd'}} \pi_m (r_m^T x_m + \sum_{k \in \mathcal{D}_m} q_{mk} \delta_{mk} + \Theta_m)$$

s.t. \( \sum_{d \in \mathcal{D}_n} \delta_{nd} = 1 \) \( n \in \mathcal{N} \), (9b)

$$A_n x_n + \sum_{d \in \mathcal{D}_n} B_{nd} \delta_{nd} + C_{a(n)} x_{a(n)} + \sum_{d \in \mathcal{D}_a(n)} D_{a(n),d} \delta_{a(n),d} = h_n$$ \( n \in \mathcal{N} \), (9c)

$$x_n \in X_{t(n)}$$ \( n \in \mathcal{N} \), (9d)

$$\delta_{nd} \in \{0, 1\}$$ \( n \in \mathcal{N}, d \in \mathcal{D}_n \), (9e)

$$\delta_{\bar{n}d'} = 1$$ (9f)

Therefore, when calculating $M_{\bar{n}d}$, the upper bound $\Theta_{\bar{n}d}^*$ in (8d) is given by:

$$\Theta_{\bar{n}d}^* = \max_{d' \in D_{\bar{n}}, d \neq \bar{d}} \Phi_{\bar{n}d'}^*$$

Clearly, solving problems (8) and (9) amounts to solving integer programs of size comparable with the original problem (7). However, the tightest $\Theta_{\bar{n}d}^*$ and $M_{\bar{n}d}$ are not necessary, and higher values would still provide correct results. A suitable value for $M_{\bar{n}d}$ can be obtained by solving any relaxation of problem (8),
yielding $M_{nd}^R$ and $\Theta_n^*$, yielding $\Phi_{nd}^R$ and in turn $\Theta_{nd}^R$. As an example, one might solve the linear programming relaxation of problems (8) and (9), or, if the size of the problems is excessively high, one might choose to relax constraints (8c) and (9c) for some stages. Finally, since the procedure outlined might return negative values for some $M_{nd}$, we set $M_{nd} = \max\{0, M_{nd}^R\}$ to reduce, when possible, high big-$M$ absolute values. The procedure is summarized in Algorithm 1.

**Algorithm 1 Algorithm for calculating $M_{nd}$ for $T - 1$**

1: Input: $\mathcal{N}, \mathcal{D}_n$ for $n \in \mathcal{N}, \Theta_n$ for $n \in \mathcal{N}_T$
2: for Node $\bar{n} \in \mathcal{N}_{T-1}$ do
3:     for Distribution $\bar{d} \in \mathcal{D}_{\bar{n}}$ do
4:         Calculate $\Phi_{\bar{n}\bar{d}}^{LP}$ by solving the LP relaxation to problem (6)
5:     end for
6: for Distribution $\bar{d} \in \mathcal{D}_{\bar{n}}$ do
7:     In constraint (8d) set $\Theta_{\bar{n}\bar{d}}^* = \max_{d \in \mathcal{D}_n : d \neq \bar{d}} \Phi_{\bar{n}\bar{d}}^{LP}$
8:     Calculate $M_{\bar{n}\bar{d}}^R$ by solving a suitable relaxation to problem (8)
9:     Set $M_{\bar{n}\bar{d}} = \max\{0, M_{\bar{n}\bar{d}}^R\}$
10: end for
11: end for
12: return $M_{nd}$ for $n \in \mathcal{N}_{T-1}$ and $d \in \mathcal{D}_n$.

C.2 Big-$M$ values for stages $t = T - 2, \ldots, 1$

Once constants $M_{nd}$ are available for every $n \in \mathcal{N}_{T-1}$ and $d \in \mathcal{D}_n$ we can proceed in a similar way to calculate big-Ms for stages $T - 2, \ldots, 1$. Given a stage $t \in \{T - 2, \ldots, 1\}$, a node at that stage, $\bar{n} \in \mathcal{N}_t$, and distribution available at that node $\bar{d} \in \mathcal{D}_{\bar{n}}$, the tightest value of constant $M_{nd}$, namely $M_{\bar{n}\bar{d}}^*$, is

$$M_{\bar{n}\bar{d}}^* = \max_{\Theta_n} \left\{ \theta_n - \sum_{m \in \mathcal{N}_{nd}} \pi_m (r_m x_m + \sum_{k \in \mathcal{D}_m} q_{mk} \theta_m) \right\} \quad (10a)$$

s.t. \[
\sum_{d \in \mathcal{D}_n} \delta_{nd} = 1 \\
A_n x_n + \sum_{d \in \mathcal{D}_n} B_{nd} \delta_{nd} + C_{\bar{a}(n)} x_{\bar{a}(n)} + \sum_{d \in \mathcal{D}_{\bar{a}(n)}} D_{a(n),d} \delta_{a(n),d} = h_n \\
\theta_n \geq \sum_{m \in \mathcal{N}_{nd}} \pi_m (r_m x_m + \sum_{k \in \mathcal{D}_m} q_{mk} \theta_m) - M_{nd}(1 - \delta_{nd}) \\
\theta_n = \Theta_n \\
\theta_n \leq \Theta_{\bar{n}\bar{d}}^* \\
x_{\bar{n}} = \mathcal{N}_{\bar{a}(n)} \\
\delta_{nd} \in \{0, 1\} \\
\delta_{\bar{n}\bar{d}} = 0 \\
\]  

(10b)  

(10c)  

(10d)  

(10e)  

(10f)  

(10g)  

(10h)  

(10i)

Problem (11) consists of finding the feasible solution to problem (7) which yields the highest value of the left-hand-side of constraint (7d) for node $\bar{n}$ and distribution $\bar{d}$. Notice that, unlike in problem (6), problem (11) includes constraints (11b) which are necessary to ensure that $\theta_n$ values are set to the lowest expectation for all stages between $t$ and $T - 1$. Note that constant $M_{nd}$ in constraints (11b) is also an upper bound to the quantity $\theta_n - \sum_{m \in \mathcal{N}_{nd}} \pi_m (r_m x_m + \sum_{k \in \mathcal{D}_m} q_{mk} \theta_m)$ and can be thus set to the quantities determined at previous iterations. Also in this case, $\Theta_{\bar{n}\bar{d}}^*$ in (11b) represents the highest possible value $\theta_n$ can take for distributions other than $\bar{d}$. The highest expectation for stage $t + 1$, given distribution $\bar{d}' \in \mathcal{D}_{\bar{n}}$ is the optimal value to problem (11)

$$\Phi_{\bar{n}\bar{d}'}^* = \max_{\Theta_n} \left\{ \theta_n - \sum_{m \in \mathcal{N}_{nd}} \pi_m (r_m x_m + \sum_{k \in \mathcal{D}_m} q_{mk} \theta_m) \right\} \quad (11a)$$

s.t. \[
\sum_{d \in \mathcal{D}_n} \delta_{nd} = 1 \\
\]  

(11b)
\[ A_n x_n + \sum_{d \in D_n} B_{nd} \delta_{nd} + C_{a(n)} x_{a(n)} + \sum_{d \in D_n} D_{a(n),d} \delta_{a(n),d} = h_n \quad n \in \mathcal{N}, \quad (11c) \]

\[ \theta_n \leq \sum_{m \in \mathcal{N}_d} \pi_m (r_m x_m + \sum_{k \in \mathcal{D}_m} q_{mk} \delta_{mk} + \theta_m) + M_{nd}(1 - \delta_{nd}) \quad t = \bar{t} + 1, \ldots, T - 1, n \in \mathcal{N}_T, d \in \mathcal{D}_n, \quad (11d) \]

\[ \theta_n = \Theta_n \quad n \in \mathcal{N}_T \quad (11e) \]

\[ x_n \in \mathcal{X}_{\mathcal{L}(n)} \quad n \in \mathcal{N}_T \quad (11f) \]

\[ \delta_{nd} \in \{0, 1\} \quad n \in \mathcal{N}, d \in \mathcal{D}_n \quad (11g) \]

\[ \delta_{\bar{n}d'} = 1 \quad (11h) \]

Therefore, the upper bound \( \Theta^*_{\bar{n}d} \) in \([10]\) is given by:

\[ \Theta^*_{\bar{n}d} = \max_{d \in \mathcal{D}_n : d \neq \bar{d}} \Phi^*_{\bar{n}d} \]

Similarly to Section \( \text{C.1} \), calculating the optimal \( M^*_{\bar{n}d} \) is cumbersome as well as not strictly necessary. Therefore, any computationally suitable relaxation to problems \([10]\) and \([11]\) can be adopted. The procedure for obtaining constants \( M_{nd} \) for stages \( T - 2, \ldots, 1 \) is sketched in Algorithm \( 2 \).

\textbf{Algorithm 2} Algorithm for calculating \( M_{nd} \) for \( \bar{t} = T - 2, \ldots, 1 \)

1: Input: \( N, D_n \) for \( n \in \mathcal{N}, \Theta_n \) for \( n \in \mathcal{N}_T, M_{nd} \) for \( n \in \mathcal{N}_T, t = \bar{t} + 1, \ldots, T - 1, d \in \mathcal{D}_n. \)
2: for Node \( \bar{n} \in \mathcal{N}_T \) do
3: for Distribution \( d \in \mathcal{D}_{\bar{n}} \) do
4: Calculate \( \Phi^L_{\bar{n}d} \) by solving the LP relaxation to problem \([11]\)
5: end for
6: for Distribution \( \bar{d} \in \mathcal{D}_{\bar{n}} \) do
7: In constraint \([10]\) set \( \Theta^*_{\bar{n}d} = \max_{d \in \mathcal{D}_n : d \neq \bar{d}} \Phi^L_{\bar{n}d} \)
8: Calculate \( M^R_{\bar{n}d} \) by solving a relaxation to problem \([10]\)
9: Set \( M_{\bar{n}\bar{d}} = \max\{0, M^R_{\bar{n}\bar{d}}\} \)
10: end for
11: end for
12: return \( M_{nd} \) for \( n \in \mathcal{N}_T \) and \( d \in \mathcal{D}_n. \)

\textbf{D} Computation time for finding big-\( M \) values for the FTCP

Table 6: Average elapsed time in seconds for the computations of big-\( M \) values using the procedure in Section \( \text{C.2} \). \( S \) indicates the number of realizations describing each distribution.

| Team | \( S \) | \( |\mathcal{L}| = 3 \) | \( |\mathcal{L}| = 4 \) | \( |\mathcal{L}| = 5 \) |
|------|------|-----------------|-----------------|-----------------|
| ARS  | 4    | 11.216          | 36.161          | 66.495          |
| ASV  | 4    | 13.387          | 37.614          | 84.259          |
| CAR  | 4    | 17.873          | 44.370          | 96.844          |
| CHE  | 4    | 16.862          | 52.053          | 149.683         |
| CRP  | 4    | 17.100          | 47.362          | 129.843         |
| EVE  | 4    | 10.921          | 36.309          | 93.248          |
| FUL  | 4    | 12.467          | 39.589          | 67.543          |
| HUL  | 4    | 11.928          | 40.945          | 86.544          |
| LIV  | 4    | 11.384          | 32.654          | 84.933          |
| MAC  | 4    | 12.999          | 32.460          | 71.190          |
| MAU  | 4    | 14.093          | 38.464          | 95.931          |
| NEC  | 4    | 11.079          | 32.661          | 87.461          |
| NOR  | 4    | 9.897           | 28.554          | 66.370          |

21
| Team | Games | SO   | TO   | Score |
|------|-------|------|------|-------|
| SOU  | 4     | 10.861 | 23.151 | 66.977 |
| STO  | 4     | 11.009 | 37.650 | 68.867 |
| SUN  | 4     | 18.335 | 36.425 | 85.658 |
| SWA  | 4     | 12.770 | 32.966 | 99.143 |
| TOT  | 4     | 11.475 | 28.322 | 69.286 |
| WBA  | 4     | 10.644 | 28.465 | 85.157 |
| WHU  | 4     | 17.390 | 45.385 | 78.934 |
| ARS  | 5     | 23.028 | 59.372 | 165.883 |
| ASV  | 5     | 21.531 | 62.066 | 167.843 |
| CAR  | 5     | 24.472 | 74.075 | 241.720 |
| CHE  | 5     | 28.144 | 84.103 | 301.168 |
| CRP  | 5     | 21.557 | 74.754 | 257.036 |
| EVE  | 5     | 21.673 | 72.323 | 183.927 |
| FUL  | 5     | 19.659 | 50.441 | 142.312 |
| HUL  | 5     | 23.749 | 51.421 | 176.955 |
| LIV  | 5     | 18.195 | 40.295 | 150.142 |
| MAC  | 5     | 20.799 | 47.659 | 117.553 |
| MAU  | 5     | 31.487 | 64.327 | 155.962 |
| NEC  | 5     | 35.180 | 64.231 | 175.313 |
| NOR  | 5     | 26.893 | 51.132 | 146.938 |
| SOU  | 5     | 18.282 | 43.805 | 120.134 |
| STO  | 5     | 21.236 | 54.687 | 151.571 |
| SUN  | 5     | 29.861 | 66.360 | 187.473 |
| SWA  | 5     | 30.952 | 59.528 | 141.202 |
| TOT  | 5     | 28.963 | 56.107 | 128.371 |
| WBA  | 5     | 23.609 | 54.868 | 139.128 |
| WHU  | 5     | 27.754 | 63.924 | 176.582 |

**Avg**     | **19.018** | **48.176** | **129.039**