An Effective Heuristic for Multistage Linear Programming with a Stochastic Right-Hand Side

C. Beltran-Royo* L. F. Escudero † J. F. Monge‡ R. E. Rodriguez-Ravines§

July 9, 2014

Abstract

The multistage Stochastic Linear Programming (SLP) problem may become numerically intractable for huge instances, in which case one can solve an approximation as for example the well known multistage Expected Value (EV) problem. We introduce a new approximation to the SLP problem that we call the multistage Event Linear Programming (ELP) problem. To obtain this approximation the SLP constraints are aggregated by means of the conditional expectation operator. Based in this new problem we derive the ELP heuristic that produces a lower and an upper bound for the SLP problem. We have assessed the validity of the ELP heuristic by solving large scale instances of the network revenue management problem, where the new approach has clearly outperformed the EV approach. One limitation of this new approach is that it only considers randomness on the right-hand side, which is assumed to be discrete and stagewise independent.

Keywords: Multistage stochastic programming, constraint aggregation, conditional expectation, scenario tree, revenue management.

1 Introduction

The Multistage Continuous Stochastic Linear Programming (MCSLP) problem corresponds to a linear programming problem with uncertainty in some of its parameters and with several decision stages. For notational simplicity, considering that all the problems here are multistage, we will drop the M and C and will use the shorter form SLP instead. The main feature of the SLP problem is that the uncertain parameters are revealed gradually over time and our decisions should be adapted to this process. The relevance, applications, properties, approaches and solution methods of this problem can be found in [4, 17, 32], among others. To address optimization problems under uncertainty one can use different approaches such as chance constraint optimization [28], robust optimization [24] and scenario based optimization [4], among others. In this paper we will focus on the last approach.

The SLP problem with a continuous stochastic process is, in general, numerically intractable. To overcome this difficulty one can approximate the original stochastic process by discretizing it. Of course, the quality of the approximation will depend on the quality of the discretization [25]. The

*Statistics and Operations Research, Rey Juan Carlos University, Madrid, Spain.
†Statistics and Operations Research, Rey Juan Carlos University, Madrid, Spain.
‡Universidad Miguel Hernandez de Elche, Spain.
§Bayes Forecast, Madrid, Spain.

discretization process consists in approximating the original stochastic process by a stochastic process with a finite support which can be represented by a scenario tree. Thus, the first difficulty in scenario based optimization corresponds to built a representative and tractable scenario tree. See [9, 15], among others. Once a representative scenario tree has been built one can write the so called deterministic equivalent problem which corresponds to a large scale structured linear programming (LP) problem.

The second difficulty corresponds to solve this LP problem. State of the art optimization software as for example IBM ILOG CPLEX Optimizer [16], or CPLEX for short, can be used to successfully solving SLP instances of moderate size. However, for many SLP instances one needs to use alternative methods which can be classified into exact and approximate ones. Exact methods can deal with a large number of scenarios and are based on Lagrangian relaxation [13], Benders decomposition [14] and interior point methods [6], among others. The performance of these optimization methods can be enhanced by using large computing systems (parallel computing, grid computing, etc.) as for example in [21]. However, if the number of scenarios becomes too large, exact methods are impractical. In this case, either one solves the SLP problem approximately or one solves an approximation to the SLP problem. This is the case of schemes such as scenario aggregation [19], scenario sampling [33], stochastic dynamic programming scenario refinement [5], approximate dynamic programming [27], and multistage stochastic decomposition [29], among others. However, even an approximated solution of the SLP problem by the Sample Average Approximation method requires an exponential number of sampled scenarios in order to attain a reasonable accuracy [31].

In [7] it is presented an iterative procedure, based on constraint aggregation, to fully solve stochastic programming problems with a convex cost and linear constraints. The idea is to reduce the number of constraints in the original problem by replacing them by aggregated constraints, which are certain weights combinations of the original ones. The method generates a sequence of problems with aggregated constraints whose iterates converge to the optimal solution set of the original problem. In contrast, the heuristic method we present is based on the so called multistage Event Linear Programming (ELP) problem which approximates the SLP one. This approximation is also based on constraint aggregation, however its objective is not to fully solve the original SLP problem but to compute a good suboptimal solution and a good cost lower bound. The second difference is that in [7] the main computations are performed in the space of the original SLP problem whose dimension may be huge, whereas in the approach we present, the main computations are performed in the space of the ELP problem whose dimension is drastically smaller than the original one. The third difference is that our approach uses the probability distribution of the stochastic parameters as the aggregate weights in contrast with [7], which uses the level of infeasibility of the current iterate for a given constraint, as the corresponding aggregate weight.

In this context it is useful to have some cost bound in order to assess the quality of the approximated solution. One of the oldest bounds is the so-called wait-and-see cost lower bound which can be obtained by solving the SLP problem without satisfying the nonanticipativity constraints [4]. Bounds based on Jensen's or on Edmundson-Madansky inequalities can be found in [4]. In [34] Jensen's bound is improved by relaxing certain constraints and associating dual multipliers with them. Another type of bound in stochastic programming is obtained by aggregation of constraints, variables or stages [2, 35]. Such approximations are shown to provide bounds if the randomness appears exclusively either in the objective or in the right-hand side (rhs). [20] constructs two discrete and stage-aggregated stochastic programs which provide upper and lower bounds on the SLP optimal cost and are numerically tractable. In the framework of the so-called sample average approximation of the SLP problem, one can infer statistical bounds for the SLP solution value as in [30].

One of the most popular approximations to the SLP problem is the multistage *Expected Value* (EV) problem, which replaces de stochastic parameters of the SLP problem by their expected value. To derive the EV problem, the scenario tree associated to the SLP problem, is reduced into a degenerate scenario tree with one scenario. That is, the EV problem approximates the SLP problem by ignoring uncertainty. As an alternative to the EV problem, we introduce the multistage Event Linear Programming (ELP) problem, which also approximates the SLP problem but without ignoring uncertainty. The ELP problem takes into account uncertainty in a simplified way: roughly speaking, it approximates the multistage scenario tree by a sequence of connected two-stage scenario trees (see Section 4). As far as we know such approach has never been proposed in literature and it could be useful in the cases where the SLP problem were numerically intractable.

In this paper we will concentrate on the SLP problem with randomness appearing exclusively in the rhs of the constraints, which is assumed to be discrete and stagewise independent. In this case, the bound given by the EV problem, corresponds to Jensen's bound. We will prove that the ELP problem also gives a cost lower bound and try to answer the following questions: Is the ELP bound tighter than the EV one? Is the ELP problem tractable? Is it possible to derive good SLP solutions by using the ELP solutions? What is the computational performance of the ELP heuristic in the case of large scale instances? To answer these questions we have used a testbed of large scale instances of the network revenue management problem (up to 393 millions of variables and 357 millions of constraints). The average CPLEX time for the EV, ELP and SLP approaches has been 198, 328 and 1995 seconds, respectively (CPLEX has failed to solve 22% of the instances when using the SLP approach). The average worst case optimality gap for the EV and ELP approaches, has been 3.62% and 0.45%, respectively.

Thus, the objectives of this paper are to introduce the ELP problem, to study some of its theoretical properties, to compare it to the EV problem, to analyze the computational effort to solve large scale ELP instances and to consider a scheme for deriving a (hopefully good) feasible solution for the SLP problem. With these objectives in mind, in Section 2 we describe the well known SLP problem and the scenario tree structure. In Section 3 we state the EV problem. In Section 4 we introduce the ELP problem and the event spike structure. In Section 5 we state and prove the theoretical results concerning the EV and the ELP bounds. In Section 6 we see an algorithm for obtaining feasible SLP solutions after solving the EV and ELP problems. Finally, in Section 7 we present the computational results of comparing the EV, ELP and SLP solution values in a large testbed of instances of the network revenue management problem, that has been chosen as the pilot case to study the effectiveness of the ELP approach.

2 The multistage LP problem with a stochastic right-hand side

The following parameters, indexes and index sets, will be used throughout the paper.

t	Index for stages, $t \in \mathcal{T} = \{1, \ldots, T\}$	
k	Index for groups of nodes of the scenario tree, $k \in \mathcal{K}_t = \{1, \ldots, K_t\}$	$\forall\, t \in \mathcal{T}$
	Two nodes are in the same group k if they have the same ancestor node (see Section 2.1)	
l	Index for nodes within the same group of nodes, $l \in \mathcal{L}_t = \{1, \ldots, L_t\}$ (see Section 2.1)	$\forall\, t \in \mathcal{T}$
$\mathcal{T}^+$	Stands for $\{2, \ldots, T\}$	
$\mathcal{T}^-$	Stands for $\{1, \ldots, T-1\}$	
$\mathcal{T}\mathcal{K}_t\mathcal{L}_t$	Stands for $\mathcal{T} \times \mathcal{K}_t \times \mathcal{L}_t$	$\forall\, t \in \mathcal{T}$

Let us consider the following multistage deterministic LP problem that we name P_{DLP}:

$$\begin{aligned} \min_x \quad & \sum_{t\in\mathcal{T}} c_t^\intercal x_t \\ \text{s.t.} \quad & A_1 x_1 = b_1 \\ & \sum_{\tau=1}^{t-1} B_{t\tau} x_\tau + A_t x_t = b_t \qquad && \forall\, t \in \mathcal{T}^+ \\ & x \geq 0, \end{aligned}$$

where c_t is the vector of the objective function coefficients, A_1 and b_1 are the constraint matrix and the right-hand side (rhs) related to stage $t = 1$. For all $t \in \mathcal{T}^+$, $B_{t\tau}$ is the constraint matrix of the decision vector x_τ related to stage $\tau < t$, A_t is the constraint matrix of the decision vector x_t and b_t is the rhs corresponding to stage t.

In real life instances, any of the parameters of P_{DLP} may be stochastic. In order to introduce our new approach, we consider a simpler stochastic version of P_{DLP} where the rhs b_t is the only random vector. A stochastic rhs typically reflects uncertainty in supply and/or demand. This is very often the case for problems arising in manufacturing, telecommunications, transportation and power generation [3, 26]. Another important example is the revenue management problem [1, 23] which we have chosen as a pilot case for our computational experience (see Section 7).

To formulate the stochastic rhs version of problem P_{DLP}, we will replace the set of deterministic vectors $\{b_t\}_{t\in\mathcal{T}}$ by the multivariate stochastic process $\{\xi_t\}_{t\in\mathcal{T}}$. On the other hand, the sequential structure of the problem implies that the decision at stage t must be contingent to the random history $\xi_{[t]} := (\xi_1, \ldots, \xi_t)$. Therefore, decision x_t is a function of $\xi_{[t]}$ and we will write $x_t(\xi_{[t]})$. For the remaining of the paper we make the following assumptions.

Assumption 1 *The multivariate stochastic process $\{\xi_t\}_{t\in\mathcal{T}}$ has the following features:*

1. *Its first component, i.e., vector ξ_1, is deterministic, being b_1.*
2. *Each random vector ξ_t has a finite support S_{ξ_t}, for all $t \in \mathcal{T}$.*
3. *It is stagewise independent, that is, the random vectors ξ_t and $\xi_{t'}$ are independent, for any $t, t' \in \mathcal{T}$ with $t \neq t'$.*
4. *It does not depend on the decision sequence $\{x_t\}_{t\in\mathcal{T}}$.*

Notice that the previous assumption implies that the support $S_{\xi_{[t]}} = S_{\xi_1} \times \ldots \times S_{\xi_t}$ is also finite.

The multistage Stochastic Linear Programming (SLP) problem corresponds to the stochastic rhs version of P_{DLP}, which can be written as:

$$\min_x \quad z_{SLP}(x) = \sum_{t\in\mathcal{T}} \mathbb{E}[\, c_t^\intercal x_t(\xi_{[t]}) \,] \tag{1}$$

$$\text{s.t.} \quad A_1 x_t(\xi_{[1]})] = \xi_1 \tag{2}$$

$$\sum_{\tau=1}^{t-1} B_{t\tau} x_\tau(\xi_{[\tau]}) + A_t x_t(\xi_{[t]}) = \xi_t \qquad \text{a.s.}, \forall\, t \in \mathcal{T}^+ \tag{3}$$

$$x_t(\xi_{[t]}) \geq 0 \qquad \text{a.s.}, \forall\, t \in \mathcal{T} \tag{4}$$

where ξ_1 is a deterministic vector. Notice that it is required that constraints (3) and (4) hold *almost surely* (a.s.), which means that they must hold with probability one [4]. Furthermore, $x_t(\xi_{[t]})$, for all $t \in \mathcal{T}$, is called a *policy* and represents the decision to be taken at stage t as a function of the random history $\xi_{[t]}$ (see for example [32]).

2.1 The multistage scenario tree

Model (1)-(4) of the SLP problem is handy for theoretical aspects but it is not appropriate for computational purposes. For this reason, in the next section we introduce the related deterministic equivalent problem, based on the scenario tree structure, which is the model to be used for computational purposes. In this section we describe the multistage scenario tree structure.

By Assumption 1 each random vector ξ_t has a finite support S_{ξ_t} of cardinality, say L_t, such that

$$S_{\xi_t} = \{\widetilde{\xi}_{t1}, \ldots, \widetilde{\xi}_{tL_t}\} \qquad \forall\, t \in \mathcal{T}.$$

Let the probability of each realization $\widetilde{\xi}_t$ be denoted

$$\pi_{tl} = P\left(\xi_t = \widetilde{\xi}_{tl}\right) \qquad \forall\, tl \in \mathcal{T}\mathcal{L}_t.$$

Given that $\{\xi_t\}_{t\in\mathcal{T}}$ has a finite support it can be represented by a multistage scenario tree, where each node at stage $t \in \mathcal{T}$ corresponds to a realization $\widetilde{\xi}_{[t]}$ of the random history $\xi_{[t]}$, and viceversa. If the stochastic process is stagewise independent, one consequence is that at a given stage t, all the nodes have the same number of successors (L_t) and the realizations of the rhs at those nodes do not depend on the realizations at their predecessor nodes. In this case, the total number of scenarios is given by the product $L_2 L_3 \cdots L_T$. Furthermore, if the number of successors is constant in all the stages, then we have L^{T-1} scenarios. In practical applications, in order to have a tractable and representative multistage scenario tree, one usually considers a reduced set of scenarios, by using scenario reduction schemes.

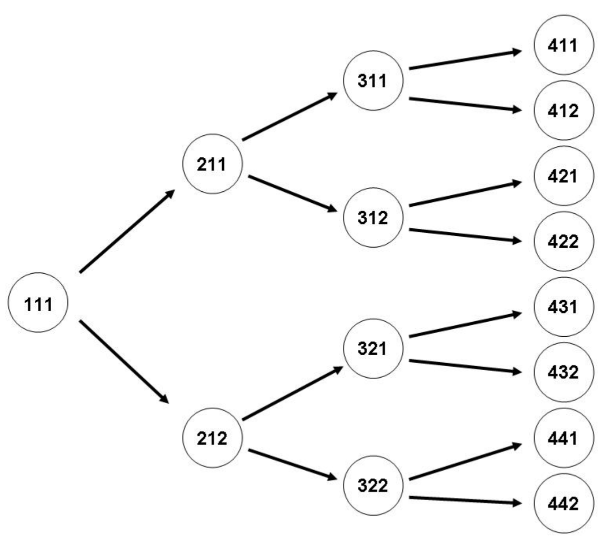

Figure 1: The scenario tree nodes are labeled by three indexes (tkl : stage t, group k and leaf l).

In this context we call ‘group of nodes’ each set of nodes with the same ancestor. We find it convenient to label the nodes of the multistage scenario tree by using three indexes: tkl, where t is the stage, k is the group of nodes and l is the node in group k. For example, in Fig. 1 we observe that at stage 2 we have only one group of nodes (node 111 is the common ancestor) and at stage 3 we have two groups of nodes. At each stage $t \in \mathcal{T}$ the groups of nodes are labeled by $k \in \mathcal{K}_t$. At each group k of stage t the nodes are labeled by $l \in \mathcal{L}_t$. This notation is not the standard one but we use it since it will be useful to compare the ELP and SLP approaches. Note that $K_t L_t$ is the number of scenario groups at stage t in the standard notation for multistage trees.

On the other hand, the vectors and the parameters associated to each node are indexed accordingly. Associated to each node tkl of the multistage scenario tree there are the following elements:

- $\widetilde{\xi}_{tl}$, realization of the random vector ξ_t. Note that under the assumption of stagewise independence we have that $\widetilde{\xi}_{tkl} = \widetilde{\xi}_{tk'l}$ for any $k, k' \in \mathcal{K}_t$ and therefore we can drop the index k and write $\widetilde{\xi}_{tl}$.
- $\widetilde{\xi}_{[t]}$, realization of the random history $\xi_{[t]}$, where
$$\widetilde{\xi}_{[t]} = (\widetilde{\xi}_{1,l(1)}, \ldots, \widetilde{\xi}_{t-1,l(t-1)}, \widetilde{\xi}_{tl})$$
is the realization of $\xi_{[t]}$ that determines node tkl.
- p_{tkl}, probability of reaching node tkl, that is,
$$p_{tkl} = P\left(\xi_{[t]} = \widetilde{\xi}_{[t]}\right) = \pi_{1,l(1)} \cdots \pi_{t-1,l(t-1)}\, \pi_{tl}.$$
For each stage $t \in \mathcal{T}$, we have that $\sum_{kl \in \mathcal{K}_t \mathcal{L}_t} p_{tkl} = 1$.
- x_{tkl}, decision vector taken at stage t after observing $\widetilde{\xi}_{[t]}$, that is,
$$x_{tkl} = x_t(\widetilde{\xi}_{[t]}).$$

2.2 Deterministic equivalent SLP problem

Based on the multistage scenario tree, the deterministic equivalent SLP problem can be written as follows:

$$\min_x \quad z_{SLP}(x) = \sum_{t \in \mathcal{T}} \sum_{kl \in \mathcal{K}_t \mathcal{L}_t} p_{tkl}\, c_t^{\intercal} x_{tkl} \tag{5}$$

$$\text{s.t.} \quad A_1 x_{111} = \widetilde{\xi}_{11} \tag{6}$$

$$\sum_{\tau=1}^{t-1} B_{t\tau} x_{a^{t-\tau}(tkl)} + A_t x_{tkl} = \widetilde{\xi}_{tl} \qquad \forall\, tkl \in \mathcal{T}^+ \mathcal{K}_t \mathcal{L}_t \tag{7}$$

$$x \geq 0, \tag{8}$$

where $\widetilde{\xi}_{11} = b_1$, $a(tkl)$ is the ancestor of node tkl, $a^2(tkl) = a(a(tkl))$ is the second ancestor of node tkl and so on. Note that in (7), the rhs term $\widetilde{\xi}_{tl}$ does not depend on the group of nodes k, and for this reason we do not write $\widetilde{\xi}_{tkl}$. See in [11] some approaches for computing or estimating the value of the SLP solution based on decomposition schemes.

3 The multistage expected value LP problem

If the SLP problem becomes too difficult, one can use the popular multistage Expected Value (EV) problem, which lower bounds the SLP problem by using the expected values of the random process instead of the random process itself. In our context, the EV problem is as follows:

$$\min_{\bar{x}} \quad z_{EV}(\bar{x}) = \sum_{t \in \mathcal{T}} c_t^\intercal \bar{x}_t \tag{9}$$

$$\text{s.t.} \quad A_1 \bar{x}_1 = \bar{\xi}_1 \tag{10}$$

$$\sum_{\tau=1}^{t-1} B_{t\tau} \bar{x}_\tau + A_t \, \bar{x}_t = \bar{\xi}_t \qquad \forall\, t \in \mathcal{T}^+ \tag{11}$$

$$\bar{x} \geq 0, \tag{12}$$

where $\bar{\xi}_t = \mathbb{E}[\xi_t]$. Notice that, since ξ_1 is a deterministic vector equal to b_1, in (10) we could just write ξ_1 instead of $\bar{\xi}_1$. However, to maintain the same notation in all the stages, we prefer to write $\bar{\xi}_1$ (in this way ξ_1 is implicitly considered a degenerate random vector which only takes a single value). See in [22] an approach for analyzing the quality of the EV solution. As we will see in Section 5.1, the decision vector $\bar{x}_t$, for all $t \in \mathcal{T}$, can be interpreted as

$$\bar{x}_t = \sum_{t \in \mathcal{T}} \sum_{kl \in \mathcal{K}_t \mathcal{L}_t} p_{tkl} x_{tkl} = \mathbb{E}[\, x_t(\xi_{[t]}) \,].$$

That is, the *resulting vector of applying the expectation operator* to the policy $x_t(\xi_{[t]})$. Notice that to compute an optimal solution $\bar{x}_t^*$ it is not necessary to know the set of optimal decisions $\{x_{tkl}^*\}_{tkl \in \mathcal{T}\mathcal{K}\mathcal{L}}$. It is enough to solve the EV problem (9)-(12).

4 The multistage event LP problem

Throughout this paper it will be used the term event, which is defined next.

Definition 1 ***Event:*** *Given a multivariate stochastic process $\{\xi_t\}_{t \in \mathcal{T}}$, let us define event as each realization $\widetilde{\xi}_{tl}$ of the random vector ξ_t, for any $t \in \mathcal{T}$.*

Notice that each scenario corresponds to a realization of the random process $\{\xi_t\}_{t \in \mathcal{T}}$. The EV problem approximates the SLP problem by ignoring uncertainty. As an alternative to the EV problem, we introduce the multistage Event Linear Programming (ELP) problem, which also approximates the SLP problem but without ignoring uncertainty. The ELP problem takes into account uncertainty in a simplified way: Instead of considering scenarios it only considers events (of course, the complete representation of the uncertainty requires scenarios). To illustrate this idea, let us consider a four-stage problem where at each future stage only two events can occur: A_t and B_t, for $t = 2, 3, 4$. Problem SLP, based on scenarios, considers all possible paths from node A_1 to nodes A_4 and B_4 to built the scenario tree (Fig. 2). In contrast, problem ELP, based on events, ignores these paths and only considers a sequence of $T - 1$ connected two-stage trees (Fig. 3). For each $t \in \mathcal{T}^-$, there is a two-stage tree whose root node accounts for the expected decision at stage t and whose leave nodes account for the possible events at stage $t + 1$ and the corresponding decisions.

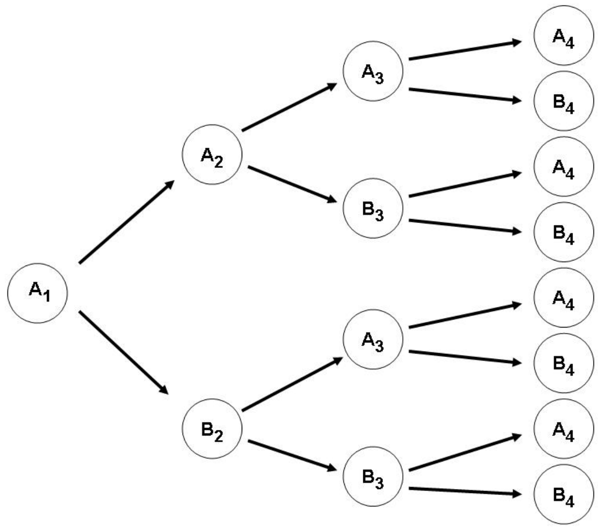

Figure 2: The 8 possible paths from A_1 to A_4 and B_4 (scenarios) can be represented by a multistage scenario tree.

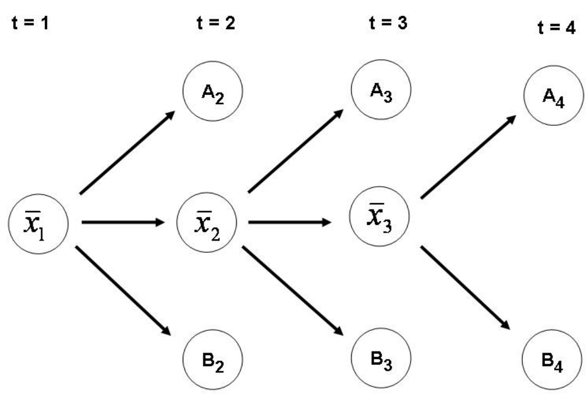

Figure 3: The scenario tree in Figure 2 can be 'approximated' by the event spike: a sequence of 3 connected two-stage trees with 3 nodes each ($\bar{x}_t$, A_{t+1} and B_{t+1}, for $t = 1, 2, 3$).

4.1 The event spike

In the case of the ELP problem, as it was the case for the SLP problem, it is convenient to consider the deterministic equivalent ELP problem for computational purposes. First, we will present the deterministic equivalent ELP problem, which is based on the event spike structure that we describe in this section. Second, we will introduce the ELP problem.

In the ELP approach, the scenario tree is approximated by the event spike, where we have the stem and one group of leaves per each stage. On the one hand, at each stage t there is one leaf node l per each event $\widetilde{\xi}_{tl}$. On the other hand, the stem corresponds to the sequence of expected decisions $\bar{x}_1, \ldots, \bar{x}_{T-1}$ (Fig. 3). Therefore, the event spike structure has two types of nodes: stem nodes labeled by the index $t \in \mathcal{T}^-$ and leaf nodes labeled by the pair $tl \in \mathcal{T}^+\mathcal{L}_t$. Note that the number of stem nodes is $T-1$ and the number of leaf nodes is $L_2 + \ldots + L_T$. If the number of leaf nodes is constant for all the stages, then the number of leaf nodes is $(T-1)L$, in contrast with the exponential number of nodes in the multistage scenario tree of our consideration. Associated to each leaf node tl there are three elements:

- $\widetilde{\xi}_{tl}$, event tl.
- π_{tl}, probability of event tl. For each stage $t \in \mathcal{T}$, we have that $\sum_{l \in \mathcal{L}_t} \pi_{tl} = 1$.
- $\hat{x}_{tl}$, decision vector.

Finally, the expected decision at stage t can be expressed as

$$\bar{x}_t = \sum_{l \in \mathcal{L}_t} \pi_{tl}\, \hat{x}_{tl} \qquad \forall\, t \in \mathcal{T}^-.$$

4.2 Deterministic equivalent ELP problem

Based on the event spike, the deterministic equivalent ELP problem can be written as follows:

$$\min_{\hat{x},\, \bar{x}} \quad z_{ELP}(\hat{x}) = \sum_{t \in \mathcal{T}} \sum_{l \in \mathcal{L}_t} \pi_{tl}\, c_t^\intercal \hat{x}_{tl} \tag{13}$$

$$\text{s.t.} \quad A_1 \hat{x}_{11} = \widetilde{\xi}_{11} \tag{14}$$

$$\sum_{\tau=1}^{t-1} B_{t\tau} \bar{x}_\tau + A_t \hat{x}_{tl} = \widetilde{\xi}_{tl} \qquad \forall\, tl \in \mathcal{T}^+\mathcal{L}_t \tag{15}$$

$$\bar{x}_t = \sum_{l \in \mathcal{L}_t} \pi_{tl}\, \hat{x}_{tl} \qquad \forall\, t \in \mathcal{T}^- \tag{16}$$

$$\hat{x} \geq 0,\ \bar{x} \geq 0. \tag{17}$$

In the previous section we have seen that $\bar{x}_t$ can be interpreted as $\mathbb{E}[x_t(\xi_{[t]})]$. In Section 5.1 we will see that the decision vector $\hat{x}_{tl}$ can be interpreted in terms of the conditional expectation such that

$$\hat{x}_{tl} = \mathbb{E}[\, x_t(\xi_{[t]}) \mid \widetilde{\xi}_{tl} \,] \qquad \forall\, tl \in \mathcal{T}\mathcal{L}_t.$$

That is, the resulting vector of applying the conditional expectation operator to the policy $x_t(\xi_{[t]})$ given that $\xi_t = \widetilde{\xi}_{tl}$.

4.3 ELP problem

The deterministic equivalent ELP problem (13)-(17) can be written in a more compact form in what we call the ELP problem:

$$\min_{\hat{x},\,\bar{x}} \quad z_{ELP}(\hat{x}) = \sum_{t\in\mathcal{T}} \mathbb{E}[\, c_t^\intercal \hat{x}_t(\xi_t)\,]$$

$$\text{s.t.} \quad A_1\hat{x}_1(\xi_1) = \xi_1 \tag{18}$$

$$\sum_{\tau=1}^{t-1} B_{t\tau}\bar{x}_\tau + A_t\hat{x}_t(\xi_t) = \xi_t \qquad \text{a.s.}, \forall\, t\in\mathcal{T}^+ \tag{19}$$

$$\bar{x}_t = \mathbb{E}[\,\hat{x}_t(\xi_t)\,] \qquad \forall\, t\in\mathcal{T}^- \tag{20}$$

$$\hat{x}_t(\xi_t) \geq 0 \qquad \text{a.s.}, \forall\, t\in\mathcal{T} \tag{21}$$

$$\bar{x}_t \geq 0 \qquad \forall\, t\in\mathcal{T}^-. \tag{22}$$

Notice that the main difference with the SLP problem corresponds to the dynamic constraint system (3). In the ELP problem, this set of constraints is approximated at each stage t by considering not the past decisions $x_1,\dots,x_{t-1}$ but its expectation $\bar{x}_1,\dots,\bar{x}_{t-1}$ as it can be seen in (19). Furthermore, at each stage t, the SLP problem considers $x_t(\xi_{[t]})$, that is, one decision vector per each realization of $\xi_{[t]}$, also known as scenario group (see e.g. [4] for the related definition in the multistage scenario tree). In contrast, at each stage t, the ELP problem only considers $\hat{x}_t(\xi_t)$, that is, one decision vector per realization of ξ_t as it can be seen in (19). Therefore, the SLP problem takes into account the random histories $\xi_{[t]}$, whereas the ELP problem only takes into account the random vectors ξ_t. Going further in the simplification, the EV problem just takes into account the expectations $\bar{\xi}_t$. In this way, the ELP and EV problems are approximations to the SLP problem that, obviously, reduce both the problem size and the stochastic accuracy. Thus, regarding tractability and 'level of stochastic information' the ELP problem lies between the SLP and the EV ones.

5 Lower bounds for the SLP solution value

5.1 Preliminary concepts and results

Let us review some preliminary concepts and results to be used in this section. First, we review the following results on *conditional expectation* [8].

Theorem 1 *([8], Theorem 4.7.1) Law of Total Probability for Expectations. Let X and Y be random variables such that Y has finite mean. Then*

$$\mathbb{E}[\,\mathbb{E}[\,Y \mid X\,]\,] = \mathbb{E}[\,Y\,],$$

where $\mathbb{E}[\,\cdot\,]$ and $\mathbb{E}[\,\cdot \mid X]$ are the expectation operator and the conditional expectation operator, respectively.

As pointed out in [8], the previous theorem also implies that for two arbitrary random variables X and Y and for an arbitrary function r such that $r(X,Y)$ has finite mean, we have

$$\mathbb{E}[\,\mathbb{E}[\,r(X,Y) \mid X\,]\,] = \mathbb{E}[\,r(X,Y)\,],$$

by letting $Z = r(X,Y)$ and noting that $\mathbb{E}[\,\mathbb{E}[\,Z \mid X\,]\,] = \mathbb{E}[\,Z\,]$. Furthermore, we can define the conditional expectation of a function $r(X_1,\dots,X_n)$ of several random variables given one or more of the variables $X_1, \dots, X_n$. In this paper we will use the following result, which generalizes Theorem 1 and can be proved in a similar way.

Theorem 2 *Let $\xi_1, \dots, \xi_n$ be random vectors and*

$$r(\xi_1, \dots, \xi_n) = (r_1(\xi_1, \dots, \xi_n), \dots, r_m(\xi_1, \dots, \xi_n))$$

be an arbitrary vectorial function of the random vectors, such that $r(\xi_1, \dots, \xi_n)$ has finite mean. Then, for any $j \in \{1, \dots, n\}$ we have that

$$\mathbb{E}[\, \mathbb{E}[\, r(\xi_1, \dots, \xi_n) \mid \xi_j \,] \,] = \mathbb{E}[\, r(\xi_1, \dots, \xi_n) \,].$$

Second, to illustrate the constraint aggregation concept [2] let us consider the following optimization problem P :

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & A_i x = b_i \qquad\qquad \forall\, i \in \mathcal{I}, \end{aligned}$$

where $\mathcal{I} = \{1, \dots, I\}$, A_i is an $m \times n$ matrix and b_i is an m-vector, for all $i \in \mathcal{I}$. Notice that there are I matrix constraints. In this context, constraint aggregation means that a set of equalities and/or inequalities is replaced by a linear combination of them. For simplicity of exposition we will assume that all the matrix constraints in problem P will be aggregated into one single matrix constraint. In general the scalars or weights of the linear combination are taken positive in order to preserve the orientation of the constraint inequalities (if any). We denote these weights by vector $\alpha = (\alpha_1, \dots, \alpha_I)$. The problem obtained by aggregating the matrix constraints, which we name P_α, corresponds to

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}} \alpha_i A_i x = \sum_{i \in \mathcal{I}} \alpha_i b_i. \end{aligned}$$

It is clear that if a solution x is feasible for problem P then it is also feasible for problem P_α. So, problem P_α is a relaxation of problem P obtained by constraint aggregation.

Third, to illustrate the concept of 'constraint aggregation induced by the expectation operator', let us consider a random vector ξ with finite support $S_\xi = \{\widetilde{\xi_i}\}_{i \in \mathcal{I}}$ and probability weights $\pi_i = P\left(\xi = \widetilde{\xi_i}\right)$ for all $i \in \mathcal{I}$. We name P^ξ the following stochastic optimization problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x(\widetilde{\xi_1}) \dots, x(\widetilde{\xi_I})) \\ \text{s.t.} \quad & Ax(\widetilde{\xi_1}) = \widetilde{\xi_1} \\ & \dots \\ & Ax(\widetilde{\xi_I}) = \widetilde{\xi_I}, \end{aligned}$$

where by abuse of notation, $x \in \mathbb{R}^n$ means $x(\widetilde{\xi_i}) \in \mathbb{R}^n$ for all $i \in \mathcal{I}$. In this context, the constraint aggregation problem P^ξ_π corresponds to

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x(\xi)) \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}} \pi_i A x(\widetilde{\xi_i}) = \sum_{i \in \mathcal{I}} \pi_i \widetilde{\xi_i}, \end{aligned}$$

which can also be written in a more compact form by using the expectation operator

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x(\xi)) \\ \text{s.t.} \quad & \mathbb{E}[\, Ax(\xi)\,] = \mathbb{E}[\, \xi\,]. \end{aligned}$$

Thus, we say that problem P_π^ξ has been derived from problem P^ξ by the constraint aggregation induced by the expectation operator $\mathbb{E}[\,\cdot\,]$. Furthermore, it is also clear that problem P_π^ξ is a relaxation of problem P^ξ obtained by constraint aggregation.

Fourth, to illustrate the use of the conditional expectation operator in this context, let us consider, additionally to the previous random vector ξ, a new random vector ζ with finite support $S_\zeta = \{\widetilde{\zeta}_j\}_{j \in \mathcal{J}}$ with $\mathcal{J} = \{1, \ldots, J\}$ and the random vector $\eta = \binom{\xi}{\zeta}$ with finite support S_η. We also consider the conditional random vectors $(\eta \mid \widetilde{\zeta}_1), \ldots, (\eta \mid \widetilde{\zeta}_J)$, which represent the random vector η given we know ζ is equal to $\widetilde{\zeta}_1$ or $\widetilde{\zeta}_2, \ldots$ or $\widetilde{\zeta}_J$, respectively. Each random vector $(\eta \mid \widetilde{\zeta}_j)$, for all $j \in \mathcal{J}$, has finite support and conditional probability weights $p_{ij} = P\left(\xi = \widetilde{\xi}_i \mid \zeta = \widetilde{\zeta}_j\right)$ for all $i \in \mathcal{I}$.

We name P^η the following stochastic optimization problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x(\eta)) \\ \text{s.t.} \quad & Ax(\eta) = \eta && \text{a.s.} \end{aligned}$$

which can be written as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x(\eta)) \\ \text{s.t.} \quad & Ax(\eta \mid \widetilde{\zeta}_1) = (\eta \mid \widetilde{\zeta}_1) && \text{a.s.} \\ & \cdots \\ & Ax(\eta \mid \widetilde{\zeta}_J) = (\eta \mid \widetilde{\zeta}_J) && \text{a.s.} \end{aligned}$$

Notice that in this problem the only random component of $(\eta \mid \widetilde{\zeta}_j)$ corresponds to the random vector ξ, since by definition $(\eta \mid \widetilde{\zeta}_j) = \binom{\xi}{\widetilde{\zeta}_j}$ for all $j \in \mathcal{J}$.

In this context, the constraint aggregation problem P_p^η corresponds to

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x(\eta)) \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}} p_{i1} Ax(\widetilde{\xi}_i, \widetilde{\zeta}_1) = \sum_{i \in \mathcal{I}} p_{i1} \begin{pmatrix} \widetilde{\xi}_i \\ \widetilde{\zeta}_1 \end{pmatrix} \\ & \cdots \\ & \sum_{i \in \mathcal{I}} p_{iJ} Ax(\widetilde{\xi}_i, \widetilde{\zeta}_J) = \sum_{i \in \mathcal{I}} p_{iJ} \begin{pmatrix} \widetilde{\xi}_i \\ \widetilde{\zeta}_J \end{pmatrix}, \end{aligned}$$

which can be written in a more compact form by using the conditional expectation operator

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x(\eta)) \\ \text{s.t.} \quad & \mathbb{E}[\, Ax(\eta) \mid \widetilde{\zeta}_1\,] = \mathbb{E}[\, \eta \mid \widetilde{\zeta}_1\,] \\ & \cdots \\ & \mathbb{E}[\, Ax(\eta) \mid \widetilde{\zeta}_J\,] = \mathbb{E}[\, \eta \mid \widetilde{\zeta}_J\,], \end{aligned}$$

or even in a more compact form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x(\eta)) \\ \text{s.t.} \quad & \mathbb{E}[\, Ax(\eta) \mid \zeta \,] = \mathbb{E}[\, \eta \mid \zeta \,] \qquad \text{a.s.} \end{aligned}$$

Notice that in this problem, after applying the conditional expectation $\mathbb{E}[\,\cdot \mid \zeta\,]$, the only random component corresponds to the random vector ζ.

In this context, we say that problem P_p^η has been derived from problem P^η by the constraint aggregation induced by the conditional expectation operator $\mathbb{E}[\,\cdot \mid \zeta\,]$. As in the previous cases, problem P_p^η is a relaxation of problem P^η obtained by constraint aggregation.

Fifth, we summarize the notation regarding the expectation operator applied to a policy $x_t(\xi_{[t]})$. For all $t \in \mathcal{T}$, the following notation is considered:

- $\mathbb{E}[\, x_t(\xi_{[t]}) \,]$: the resultant decision vector of applying the expectation operator to the policy $x_t(\xi_{[t]})$. Vector $\mathbb{E}[\, x_t(\xi_{[t]}) \,]$ corresponds to vector $\bar{x}_t$ in the EV problem.
- $\mathbb{E}[\, x_t(\xi_{[t]}) \mid \xi_t \,]$: the resultant policy of applying the conditional expectation operator to the policy $x_t(\xi_{[t]})$ given ξ_t.
- $\hat{x}_t(\xi_t)$: A shorter expression for the policy $\mathbb{E}[\, x_t(\xi_{[t]}) \mid \xi_t \,]$. Notice that a policy $\hat{x}_t(\xi_t)$ in the ELP problem is a function of the random vector ξ_t whereas a policy $x_t(\xi_{[t]})$ in the SLP problem is a function of the random history $\xi_{[t]}$. Vector $\hat{x}_t(\widetilde{\xi}_{tl})$ corresponds to vector $\hat{x}_{tl}$ in the deterministic equivalent ELP problem.

5.2 Lower bounds

It is well-known that for the multistage Stochastic Linear Programming (SLP) problem with the uncertainty only in the right-hand side (rhs), the multistage Expected Value (EV) problem gives a lower bound for the SLP solution value [2]. In the context of this paper it means:

$$z^*_{EV} \leq z^*_{SLP},$$

where z^*_P stands for the solution value of a given problem P. In this section we will prove that the multistage Event Linear Programming (ELP) problem gives a stronger bound, that is, it will be proved that

$$z^*_{EV} \leq z^*_{ELP} \leq z^*_{SLP}.$$

In Theorem 3, it will be proved that $z^*_{ELP} \leq z^*_{SLP}$. It is worthy to mention that, the proof of Theorem 3 will also show the connection between problems SLP and ELP: roughly speaking the ELP problem can be derived by the constraint aggregation induced by the conditional expectation operator applied to the SLP constraints.

Theorem 3 *Let us consider the SLP problem with the rhs defined by the multivariate stochastic process $\{\xi_t\}_{t \in \mathcal{T}}$. If this stochastic process is stagewise independent then*

$$z^*_{ELP} \leq z^*_{SLP}. \tag{23}$$

Proof: The proof consists of nine steps.

First, constraint (3) of the SLP problem can be aggregated by using the conditional expectation operator $\mathbb{E}[\,\cdot \mid \xi_t]$:

$$\mathbb{E}[\sum_{\tau=1}^{t-1} B_{t\tau} x_\tau(\xi_{[\tau]})) + A_t x_t(\xi_{[t]}) \mid \xi_t\,] = \mathbb{E}[\,\xi_t \mid \xi_t\,] \qquad \text{a.s.}, \forall\, t \in \mathcal{T}^+$$

$$\Updownarrow$$

$$\sum_{\tau=1}^{t-1} B_{t\tau}\mathbb{E}[\,x_\tau(\xi_{[\tau]}) \mid \xi_t\,] + A_t\mathbb{E}[\,x_t(\xi_{[t]}) \mid \xi_t\,] = \xi_t \qquad \text{a.s.}, \forall\, t \in \mathcal{T}^+$$

$$\Updownarrow$$

$$\sum_{\tau=1}^{t-1} B_{t\tau}\mathbb{E}[\,x_\tau(\xi_{[\tau]})\,] + A_t\mathbb{E}[\,x_t(\xi_{[t]}) \mid \xi_t\,] = \xi_t \qquad \text{a.s.}, \forall\, t \in \mathcal{T}^+, \tag{24}$$

where, to deduce the last equation we have used that by hypothesis the stochastic process $\{\xi_t\}_{t\in\mathcal{T}}$ is stagewise independent.

Now, considering that $x_t(\xi_{[t]})$ is a vectorial function of the random vectors $\xi_1, \ldots, \xi_t$, by Theorem 2 we can write

$$\mathbb{E}[\,x_t(\xi_{[t]})\,] = \mathbb{E}[\,\mathbb{E}[\,x_t(\xi_{[t]}) \mid \xi_t\,]\,] \qquad \forall\, t \in \mathcal{T}.$$

By using this equality, equation (24) can be rewritten as

$$\sum_{\tau=1}^{t-1} B_{t\tau}\mathbb{E}[\,\mathbb{E}[\,x_\tau(\xi_{[\tau]}) \mid \xi_\tau\,]\,] + A_t\mathbb{E}[\,x_t(\xi_{[t]}) \mid \xi_t\,] = \xi_t \qquad \text{a.s.}, \forall\, t \in \mathcal{T}^+.$$

Second, constraint (2) of the SLP problem is related to the first stage and it is equivalent to

$$A_1\mathbb{E}[\,x_1(\xi_{[1]}) \mid \xi_1\,] = \xi_1.$$

Third, constraint (4) of the SLP problem can also be aggregated by using the conditional expectation operator:

$$\mathbb{E}[\,x_t(\xi_{[t]}) \mid \xi_t\,] \geq 0 \qquad \text{a.s.}, \forall\, t \in \mathcal{T}.$$

Fourth, by Theorem 2, the SLP objective function can be rewritten as

$$\begin{aligned} z_{SLP}(x) &= \sum_{t\in\mathcal{T}} \mathbb{E}[c_t^\intercal x_t(\xi_{[t]})\,] = \sum_{t\in\mathcal{T}} c_t^\intercal \mathbb{E}[x_t(\xi_{[t]})\,] \\ &= \sum_{t\in\mathcal{T}} c_t^\intercal \mathbb{E}[\,\mathbb{E}[\,x_t(\xi_{[t]}) \mid \xi_t\,]\,] = \sum_{t\in\mathcal{T}} \mathbb{E}[\,c_t^\intercal \mathbb{E}[\,x_t(\xi_{[t]}) \mid \xi_t\,]\,]. \end{aligned}$$

Fifth, after applying the conditional expectation operator to the SLP problem, it results the following problem with aggregated constraints which is named P_1 :

$$\begin{aligned} \min_x \quad & z_{P_1}(x) = \sum_{t\in\mathcal{T}} \mathbb{E}[c_t^\intercal\, \mathbb{E}[\,x_t(\xi_{[t]}) \mid \xi_t\,]\,] \\ \text{s.t.} \quad & A_1\mathbb{E}[\,x_1(\xi_{[1]}) \mid \xi_1\,] = \xi_1 \\ & \sum_{\tau=1}^{t-1} B_{t\tau}\mathbb{E}[\,\mathbb{E}[\,x_\tau(\xi_{[\tau]}) \mid \xi_\tau\,]\,] + A_t\mathbb{E}[\,x_t(\xi_{[t]}) \mid \xi_t\,] = \xi_t && \text{a.s.}, \forall\, t \in \mathcal{T}^+ \\ & \mathbb{E}[\,x_t(\xi_{[t]}) \mid \xi_t\,] \geq 0 && \text{a.s.}, \forall\, t \in \mathcal{T}. \end{aligned}$$

Sixth, to proceed with the proof, problem P_2 is derived from problem P_1 as follows:

$$\min_{\hat{x}} \quad z_{P_2}(\hat{x}) = \sum_{t\in\mathcal{T}} \mathbb{E}[\, c_t^\intercal \hat{x}_t(\xi_t) \,] \tag{25}$$

$$\text{s.t.} \quad A_1\hat{x}_1(\xi_1) = \xi_1 \tag{26}$$

$$\sum_{\tau=1}^{t-1} B_{t\tau}\mathbb{E}[\, \hat{x}_\tau(\xi_\tau) \,] + A_t\hat{x}_t(\xi_t) = \xi_t \qquad \text{a.s.}, \forall\, t \in \mathcal{T}^+ \tag{27}$$

$$\hat{x}_t(\xi_t) \geq 0 \qquad \text{a.s.}, \forall\, t \in \mathcal{T}, \tag{28}$$

where vector $\hat{x}_t(\xi_t)$ accounts for $\mathbb{E}[\, x_t(\xi_{[t]}) \mid \xi_t \,]$ for all $t \in \mathcal{T}$.

Seventh, problem P_2 can be written equivalently as

$$\begin{aligned} \min_{\hat{x},\, \bar{x}} \quad & \sum_{t\in\mathcal{T}} \mathbb{E}[\, c_t^\intercal \hat{x}_t(\xi_t) \,] & \\ \text{s.t.} \quad & A_1\hat{x}_1(\xi_1) = \xi_1 & \\ & \sum_{\tau=1}^{t-1} B_{t\tau}\bar{x}_\tau + A_t\hat{x}_t(\xi_t) = \xi_t & \text{a.s.}, \forall\, t \in \mathcal{T}^+ \\ & \bar{x}_t = \mathbb{E}[\, \hat{x}_t(\xi_t) \,] & \forall\, t \in \mathcal{T}^- \\ & \hat{x}_t(\xi_t) \geq 0 & \text{a.s.}, \forall\, t \in \mathcal{T} \\ & \bar{x}_t \geq 0, & \forall\, t \in \mathcal{T}^- \end{aligned}$$

which is the ELP problem.

Eighth, by comparing the solution values of all these problems it can be written

$$z^*_{ELP} = z^*_{P_2} \leq z^*_{P_1} \leq z^*_{SLP},$$

where the first inequality on the left will be proved in step nine and the second inequality comes from the fact that problem P_1 is a relaxation of the SLP problem obtained by constraint aggregation.

Ninth, to conclude, it will be proved that $z^*_{P_2} \leq z^*_{P_1}$. Let us assume that x^* is an optimal solution for P_1 and define $\hat{x}$ such that

$$\hat{x}_t(\widetilde{\xi}_{tl}) = \mathbb{E}[\, x^*_t(\xi_{[t]}) \mid \widetilde{\xi}_{tl} \,] \qquad \forall\, \widetilde{\xi}_{tl} \in S_{\xi_t}, \forall\, t \in \mathcal{T}.$$

It is easy to see that $\hat{x}$ thus defined is feasible for problem P_2 (stated in (25)-(28)) and that $z_{P_2}(\hat{x}) = z_{P_1}(x^*)$ which proves $z^*_{P_2} \leq z^*_{P_1}$. ∎

In Theorem 4 it will be proved that $z^*_{EV} \leq z^*_{ELP}$. The proof of Theorem 4 will also show the connection between the ELP and the EV problems: roughly speaking the EV problem can be derived by the constraint aggregation induced by the expectation operator applied to the ELP constraints.

Theorem 4 *The solution value of the EV problem is a lower bound of the solution value of the ELP problem, i.e.,*

$$z^*_{EV} \leq z^*_{ELP}.$$

Proof: This proof is analogous to the proof of Theorem 3 (it also consists of nine steps).

First, constraints (19) of the ELP problem can be aggregated by using the expectation operator $\mathbb{E}[\,\cdot\,]$ for all $t \in \mathcal{T}^+$:

$$\mathbb{E}[\sum_{\tau=1}^{t-1} B_{t\tau}\bar{x}_\tau + A_t\hat{x}_t(\xi_t)\,] = \mathbb{E}[\,\xi_t\,] \qquad \forall\, t \in \mathcal{T}^+$$

$$\Updownarrow$$

$$\sum_{\tau=1}^{t-1} B_{t\tau}\bar{x}_\tau + A_t\mathbb{E}[\,\hat{x}_t(\xi_t)\,] = \bar{\xi}_t \qquad \forall\, t \in \mathcal{T}^+$$

where $\bar{\xi}_t := \mathbb{E}[\,\xi_t\,]$.

Second, constraint (18) of the ELP problem is related to the first stage and then, being a deterministic one, it is equivalent to

$$A_1\mathbb{E}[\,\hat{x}_1(\xi_1)\,] = \bar{\xi}_1.$$

Third, constraint (21) of the ELP problem can also be aggregated by using the expectation operator:

$$\mathbb{E}[\,\hat{x}_t(\xi_t)\,] \geq 0 \qquad \forall\, t \in \mathcal{T}.$$

Fourth, the ELP objective function can be rewritten as

$$\begin{aligned} z_{ELP}(x) &= \sum_{t\in\mathcal{T}} \mathbb{E}[\,c_t^\intercal \hat{x}_t(\xi_t)\,] \\ &= \sum_{t\in\mathcal{T}} c_t^\intercal \mathbb{E}[\,\hat{x}_t(\xi_t)\,] \end{aligned}$$

Fifth, after applying the expectation operator to the ELP problem, it results the following problem with aggregated constraints which is named P_3 :

$$\begin{aligned} \min_{\hat{x},\,\bar{x}} \quad & z_{P_3}(\hat{x}) = \sum_{t\in\mathcal{T}} c_t^\intercal\, \mathbb{E}[\,\hat{x}_t(\xi_t)\,] \\ \text{s.t.} \quad & A_1\mathbb{E}[\,\hat{x}_1(\xi_1)\,] = \bar{\xi}_1 \\ & \sum_{\tau=1}^{t-1} B_{t\tau}\bar{x}_\tau + A_t\mathbb{E}[\,\hat{x}_t(\xi_t)\,] = \bar{\xi}_t && \forall\, t \in \mathcal{T}^+ \\ & \bar{x}_t = \mathbb{E}[\,\hat{x}_t(\xi_t)\,] && \forall\, t \in \mathcal{T}^- \\ & \mathbb{E}[\,\hat{x}_t(\xi_t)\,] \geq 0 && \forall\, t \in \mathcal{T} \\ & \bar{x}_t \geq 0 && \forall\, t \in \mathcal{T}^-. \end{aligned}$$

Sixth, to proceed with the proof, problem P_4 is derived from problem P_3 as follows

$$\min_{\breve{x},\,\bar{x}} \quad z_{P_4}(\breve{x}) = \sum_{t\in\mathcal{T}} c_t^\intercal \breve{x}_t \tag{29}$$

$$\text{s.t.} \quad A_1\breve{x}_1 = \bar{\xi}_1 \tag{30}$$

$$\sum_{\tau=1}^{t-1} B_{t\tau}\bar{x}_\tau + A_t\breve{x}_t = \bar{\xi}_t \qquad \forall\, t \in \mathcal{T}^+ \tag{31}$$

$$\bar{x}_t = \breve{x}_t \qquad \forall\, t \in \mathcal{T}^- \tag{32}$$

$$\breve{x}_t \geq 0 \qquad \forall\, t \in \mathcal{T} \tag{33}$$

$$\bar{x}_t \geq 0 \qquad \forall\, t \in \mathcal{T}^-, \tag{34}$$

where vector $\breve{x}_t$ accounts for $\mathbb{E}[\, \hat{x}_t(\xi_t) \,]$ for all $t \in \mathcal{T}$.

Seventh, we can rewrite P_4 as

$$\begin{aligned}
\min_{\breve{x}} \quad & z_{P_4}(\breve{x}) = \sum_{t \in \mathcal{T}} c_t^{\intercal} \breve{x}_t \\
\text{s.t.} \quad & A_1 \breve{x}_1 = \bar{\xi}_1 \\
& \sum_{\tau=1}^{t-1} B_{t\tau} \breve{x}_\tau + A_t \breve{x}_t = \bar{\xi}_t && \forall\, t \in \mathcal{T}^+ \\
& \breve{x}_t \geq 0 && \forall\, t \in \mathcal{T},
\end{aligned}$$

which corresponds to the EV problem.

Eighth, by comparing the solution values of all these problems it can be written

$$z^*_{EV} = z^*_{P_4} \leq z^*_{P_3} \leq z^*_{ELP},$$

where the first inequality on the left will be proved in step nine and the second inequality comes from the fact that problem P_3 is a relaxation of the ELP problem obtained by constraint aggregation.

Ninth, to conclude, it will be proved that $z^*_{P_4} \leq z^*_{P_3}$. Let us assume that $(\hat{x}^*, \bar{x}^*)$ is an optimal solution for P_3 and define $\breve{x}$ such that

$$\breve{x}_t = \mathbb{E}[\, \hat{x}^*_t(\xi_t) \,] \qquad \forall\, t \in \mathcal{T}.$$

It is easy to see that $(\breve{x}, \bar{x}^*)$ thus defined is feasible for problem P_4 (stated in (29)-(34)) and that $z_{P_4}(\breve{x}) = z_{P_3}(\hat{x}^*)$ which proves $z^*_{P_4} \leq z^*_{P_3}$. ■

Corollary 1 *Let us consider the SLP problem with the rhs defined by the multivariate stochastic process $\{\xi_t\}_{t \in \mathcal{T}}$. If the stochastic process is stagewise independent then*

$$z^*_{EV} \leq z^*_{ELP} \leq z^*_{SLP}.$$

6 Algorithm for obtaining the expected result of using the expected value solution for multistage stochastic optimization

The methodology for obtaining the Expected result of using the Expected Value solution (EEV) is very well established in the two-stage environment, see a good exposition in e.g. [4], but it is a difficult one for multistage problems, see [22]. The main difficulty lies in the same concept of multistage EEV. Alternatively, we use the methodology introduced in [10] for obtaining EEV in a rolling horizon type of calculation. Basically, it is as follows:

1. Retain the EV solution for the first stage.
2. Once the first stage EV solution is fixed, stage $t = 2$ is considered, so that $K_t L_t$ independent scenario subtrees remain.
3. The EV solutions are independently obtained for these scenario subtrees.
4. The first stage solution of the EV problem for each scenario subtree is retained and, then, fixed.
5. The procedure continues for the other stages in the scenario tree until stage T is reached.

So, at the end of the process there is a solution for each node in the scenario tree (and, then, for each scenario). The EEV is obtained by weighting the solution values for the scenarios as they have been calculated by the procedure. Let us specify the details of this methodology in Algorithm 1.

Algorithm 1: Multistage EEV.

1. *Objective*: To obtain a good feasible solution for problem SLP given by equations (5)-(8).
2. *Initialization*:
   (a) Iteration counter $t := 0$.
   (b) Set of constraints $\mathcal{C}_1 := \emptyset$.
3. *Main iteration*: Repeat while $t < T$:
   (a) Set $t := t + 1$.
   (b) Consider problem SLP (5)-(8) with the additional constraints given by $\mathcal{C}_t$. This problem can be decomposed into $K_t L_t$ independent subproblems SLP with $T + 1 - t$ stages (one per node tkl, $\forall kl \in \mathcal{K}_t\mathcal{L}_t$).
   (c) Let $P_{SLP}(tkl)$ be the subproblem corresponding to node tkl, $\forall kl \in \mathcal{K}_t\mathcal{L}_t$.
   (d) Let $P_{EV}(tkl)$ be the EV subproblem that approximates subproblem $P_{SLP}(tkl)$, $\forall kl \in \mathcal{K}_t\mathcal{L}_t$.
   (e) Solve subproblem $P_{EV}(tkl)$ to obtain an optimal solution $\bar{x}^*(tkl)$, $\forall kl \in \mathcal{K}_t\mathcal{L}_t$.
   (f) Set $\mathcal{C}_{t+1} := \mathcal{C}_t \cup \{"x_{tkl} = \bar{x}_1^*(tkl)" \mid \forall kl \in \mathcal{K}_t\mathcal{L}_t\}$.
4. *Solution recovering*:
   (a) Set the EEV solution as $x_{tkl}^{EEV} := \bar{x}_1^*(tkl) \ \ \forall tkl \in \mathcal{T}\mathcal{K}_t\mathcal{L}_t$.
   (b) Compute the EEV value as $z_{SLP}(x^{EEV})$.

Notice that in the previous algorithm:

- It is not guaranteed that the EEV solution is feasible for problem SLP.
- However, it is not difficult to prove that, if all subproblems $P_{EV}(tkl)$ for all $tkl \in \mathcal{T}\mathcal{K}_t\mathcal{L}_t$ are feasible, then the EEV solution is feasible for problem SLP and, in this case, EEV is an upper bound to the optimal value of problem SLP. This has been the case in the revenue management instances solved in Section 7.
- We have to solve a subproblem EV per node of the SLP scenario tree, that is, the total number of subproblems EV has order L^{T-1} (assuming that the number of possible events per stage is constant and equal to L).
- At each iteration t the subproblems $P_{EV}(tkl)$, $\forall kl \in K_t L_t$, are independent and therefore can be solved in parallel. That is, Algorithm 1 is highly appropriate for parallel computing.

On the other hand, the Expected result of using the ELP solution (EELP) can be obtained analogously, that is, in the previous algorithm we should replace EV with ELP.

7 Computational experience

7.1 The network revenue management problem

For computationally assessing the validity of the ELP heuristic that combines the ELP lower bound and the EELP upper bound, we have chosen as our pilot case the so named network revenue management problem for the flight tickets selling, taken from [23]. The aim of revenue management in general consists of maximizing the revenue of selling limited quantities of a set of resources by means of demand management decisions. A resource in revenue management is usually a perishable product/service, such as seats on a single flight leg or hotel rooms for a given date. It is common in revenue management that multiple resources are sold in 'bundles'. For instance, connecting flight legs are sold on a single ticket and hotel customers may stay multiple nights. In this case, the lack of availability of any resource will prevent sales of the bundle, which creates interdependence among these resources. Consequently, the demand management decisions of these resources must be coordinated. It is usual in revenue management to protect the availability of resources along the time interval that ends at the period at which the resources are used. So, the accumulative booking of each resource is prevented beyond a protection level that is, obviously, increasing along the time horizon.

The following notation is used to describe the network revenue management problem:

Sets:

$\mathcal{H}$**,** set of resources (with size H).

$\mathcal{I}$**,** set of bundles (with size I).

$\mathcal{J}$**,** set of fare classes (with size J).

$\mathcal{I}_h$**,** set of bundles using resource h, for $h \in \mathcal{H}$.

$\mathcal{IJ}$**,** stands for $\mathcal{I} \times \mathcal{J}$.

Deterministic parameters:

f^{ij}**,** fare of bundle-class ij, for $ij \in \mathcal{IJ}$.

C^h**,** capacity on resource h, for $h \in \mathcal{H}$.

$ub(P^{ij}_{tkl})$**,** upper bound on the protection level variable P^{ij}_{tkl} (defined below) for $ij \in \mathcal{IJ}$, $tkl \in \mathcal{T}^-\mathcal{K}_t\mathcal{L}_t$.

Uncertain parameters:

ξ^{ij}_{tl}**,** demand for bundle-class ij in period t at node tkl, for $tkl \in \mathcal{T}^+\mathcal{K}_t\mathcal{L}_t$.

Variables:

b^{ij}_{tkl}**,** number of accepted bookings for bundle-class ij in period t at node tkl, for $ij \in \mathcal{IJ}$, $tkl \in \mathcal{T}^+\mathcal{K}_t\mathcal{L}_t$.

B^{ij}_{tkl}, cumulative number of accepted bookings of bundle-class ij along the path from the root to node tkl, for $ij \in \mathcal{IJ}$, $tkl \in \mathcal{T}^+\mathcal{K}_t\mathcal{L}_t$.

$P^{ij}_{a(tkl)}$, protection level of bundle-class ij set at the immediate ancestor, say $a(tkl)$ of node tkl, along the path from the root to node tkl, for $ij \in \mathcal{IJ}$, $tkl \in \mathcal{T}^+\mathcal{K}_t\mathcal{L}_t$.

SLP problem

The multistage Stochastic Linear Programming (SLP) problem for network revenue management with protection levels is as follows:

$$\max_{b,B,P} \quad \sum_{tkl \in \mathcal{T}^+\mathcal{K}_t\mathcal{L}_t} p_{tkl} \sum_{ij \in \mathcal{IJ}} f^{ij} b^{ij}_{tkl} \tag{35}$$

$$\text{s.t.} \quad B^{ij}_{tkl} = B^{ij}_{a(tkl)} + b^{ij}_{tkl} \qquad \forall\, tkl \in \mathcal{T}^+\mathcal{K}_t\mathcal{L}_t,\ ij \in \mathcal{IJ} \tag{36}$$

$$B^{ij}_{tkl} \leq P^{ij}_{a(tkl)} \qquad \forall\, tkl \in \mathcal{T}^+\mathcal{K}_t\mathcal{L}_t,\ ij \in \mathcal{IJ} \tag{37}$$

$$\sum_{ij \in \mathcal{I}_h\mathcal{J}} P^{ij}_{T-1,kl} \leq C^h \qquad \forall\, kl \in \mathcal{K}_{T-1}\mathcal{L}_{T-1},\ h \in \mathcal{H} \tag{38}$$

$$0 \leq b^{ij}_{tkl} \leq \widetilde{\xi}^{ij}_{tl} \qquad \forall\, tkl \in \mathcal{T}^+\mathcal{K}_t\mathcal{L}_t,\ ij \in \mathcal{IJ} \tag{39}$$

$$0 \leq P^{ij}_{tkl} \leq ub(P^{ij}_{tkl}) \qquad \forall\, tkl \in \mathcal{T}^-\mathcal{K}_t\mathcal{L}_t,\ ij \in \mathcal{IJ}. \tag{40}$$

Constraints (36) define the booking balance equations. Notice that B^{ij}_{111} corresponds to the initial number of accepted bookings, which is a parameter of the problem for all $ij \in \mathcal{IJ}$. Constraints (37) ensure that the cumulative number of accepted bookings along the path from the root to node tlk cannot exceed the protection level set at the ancestor node $a(tlk)$. Notice that all the nodes with the same immediate ancestor share the same protection level. The protection levels across bundles and class fares are then bounded by the capacity on the resources in constraints (38). Constraints (39) reflect that the number of accepted bookings should be not greater than the demand. Constraints (40) bound the protection levels. Notice that the non-anticipativity constraints are satisfied by construction; see e.g. in [4] good expositions of this important concept on stochastic optimization. We will refer to this problem as SLP.

EV problem

The multistage Expected Value (EV) problem is as follows:

$$\max_{\bar{b},\bar{B},\bar{P}} \quad \sum_{t \in \mathcal{T}^+} \sum_{ij \in \mathcal{IJ}} f^{ij} \bar{b}^{ij}_t \tag{41}$$

$$\text{s.t.} \quad \bar{B}^{ij}_t = \bar{B}^{ij}_{t-1} + \bar{b}^{ij}_t \qquad \forall\, t \in \mathcal{T}^+,\ ij \in \mathcal{IJ} \tag{42}$$

$$\bar{B}^{ij}_t \leq \bar{P}^{ij}_{t-1} \qquad \forall\, t \in \mathcal{T}^+,\ ij \in \mathcal{IJ} \tag{43}$$

$$\sum_{ij \in \mathcal{I}_h\mathcal{J}} \bar{P}^{ij}_{T-1} \leq C^h \qquad \forall\, h \in \mathcal{H} \tag{44}$$

$$0 \leq \bar{b}^{ij}_t \leq \bar{\xi}^{ij}_t \qquad \forall\, t \in \mathcal{T}^+,\ ij \in \mathcal{IJ} \tag{45}$$

$$0 \leq \bar{P}^{ij}_t \leq ub(\bar{P}^{ij}_t) \qquad \forall\, t \in \mathcal{T}^-,\ ij \in \mathcal{IJ}. \tag{46}$$

Notice that in the EV problem the decision vectors are denoted by $\bar{b}$, $\bar{B}$ and $\bar{P}$.

ELP problem

The multistage Event Linear Programming (ELP) problem is as follows:

$$\max_{\hat{b},\hat{B},\bar{B},\bar{P}} \quad \sum_{tl\in\mathcal{T}^+\mathcal{L}_t} \pi_{tl} \sum_{ij\in\mathcal{IJ}} f^{ij}\,\hat{b}^{ij}_{tl} \tag{47}$$

$$\text{s.t.} \quad \widehat{B}^{ij}_{tl} = \bar{B}^{ij}_{t-1} + \hat{b}^{ij}_{tl} \qquad \forall\, tl \in \mathcal{T}^+\mathcal{L}_t,\ ij \in \mathcal{IJ} \tag{48}$$

$$\widehat{B}^{ij}_{tl} \le \bar{P}^{ij}_{t-1} \qquad \forall\, tl \in \mathcal{T}^+\mathcal{L}_t,\ ij \in \mathcal{IJ} \tag{49}$$

$$\sum_{ij\in\mathcal{I}_h\mathcal{J}} \bar{P}^{ij}_{T-1} \le C^h \qquad \forall\, h \in \mathcal{H} \tag{50}$$

$$\bar{B}^{ij}_{t} = \sum_{l\in\mathcal{L}_t} \pi_{tl}\widehat{B}^{ij}_{tl} \qquad \forall\, t \in \mathcal{T}^-,\ ij \in \mathcal{IJ} \tag{51}$$

$$0 \le \hat{b}^{ij}_{tl} \le \widetilde{\xi}^{ij}_{tl} \qquad \forall\, tl \in \mathcal{T}^+\mathcal{L}_t,\ ij \in \mathcal{IJ} \tag{52}$$

$$0 \le \bar{P}^{ij}_{t} \le ub(\bar{P}^{ij}_{t}) \qquad \forall\, t \in \mathcal{T}^-,\ ij \in \mathcal{IJ}. \tag{53}$$

Notice that in the ELP problem the decision vectors are denoted by $\hat{b}$, $\hat{B}$, $\bar{B}$ and $\bar{P}$. Furthermore, $\widehat{B}^{ij}_{11}$ corresponds to the initial number of accepted bookings, which is a parameter of the problem for all $ij \in \mathcal{IJ}$.

7.2 Testbed description

In this numerical study, we consider a small network, a medium, and a large one. Next, we introduce first the dimensions and then the structure of our test networks. Finally, we discuss the fares and the demand model.

Recall that, in this context, the dimensions of a network are given by its number of resources H, bundles I and fare classes J. So, they are as follows: SMALL network: $H = 10$, $I = 18$, $J = 2$; MEDIUM network: $H = 100$, $I = 200$, $J = 2$; and LARGE network: $H = 500$, $I = 1000$, $J = 2$.

In terms of network structure, we follow the standard practice in the literature to use randomly generated networks with a hub-and-spoke structure [1], resembling networks seen in hub-based airlines. Among the resources, the first half are spoke-to-hub flight legs and the second half are hub-to-spoke flight legs. The bundles are generated as follows. In the SMALL network, the first 10 bundles each use one of the 10 legs, and the remaining 8 each use two random legs, `spoke1-to-hub` and `hub-to-spoke2`. In the MEDIUM network, the first 100 bundles each use one of the 100 legs, the next 75 each use two random legs, `spoke1-to-hub` and `hub-to-spoke2`, and finally the last 25 each use four random legs `spoke1-to-hub`, `hub-to-spoke2`, `spoke2-to-hub` and `hub-to-spoke1`. Similarly, in the LARGE network, the first 500 bundles each use one of the 500 legs, the next 375 each use two random legs, and finally the last 125 each use four random legs. See [12].

We now present the way the fares and the demands have been generated following the scheme presented in [12]. The class 1 fare of a single-resource bundle is generated from a normal distribution with mean 100 and standard deviation 40, truncated within the interval $[20, 180]$. The class 1 fare of a multi-resource bundle is the summation of the class 1 fares of the single-resource bundles associated with its resources. For all bundles, the fare of class 2 is 1.4 times that of class 1. As usual in the literature, demands are modeled using a Poisson distribution. The demand for bundle-class ij in period t is generated from a Poisson distribution with mean $\mu_{ijt} := \beta_{ijt}\,\mu$, where

$$\mu = \eta \frac{\sum_{h\in\mathcal{H}} C^h}{I \times J \times T}.$$

Since all the parameters that define μ are constant, the higher η the higher the demand. Therefore, we refer to η as the "load factor of demand", which has been set equal to 2 in this computational experience. With respect to β_{ijt}, we have chosen them higher for single-resource bundles than for multi-resource ones. These parameters are also dynamic with respect to the time-to-service. For $j = 1, 2$, β_{ijt} decreases when getting closer to the departure, eventually becoming zero. Notice that the demands thus modeled are independent and the ELP heuristic can be used. However, this simplification in the demand model may be inaccurate. Thus, for example, the opportunity cost of neglecting dependency of demands across fare classes can be of the order of 2% [18]. Therefore, further research effort should be devoted to allow for correlated demands (rhs) in the ELP heuristic as pointed out in Section 8.

We use the optimization engine CPLEX v12.3, for solving the LP problems arising. Our experiments were conducted on a PC with a 2.33 GHz Intel Xeon dual core processor, 8.5 GB of RAM and the operating system was LINUX Debian 4.0.

Tables 1, 2 and 3 present for the 20 instances for each of the three network sizes tested in our computational experience, the scenario tree structure and the dimensions of the three LP problems subject of our computational comparison testing, namely, SLP (35)-(40), EV (41)-(46) and ELP (47)-(53). The headings are as follows: $case$, instance name; T, the number of periods; L, number of immediate successor nodes (obviously, belonging to the next stage) from a node of any stage; N, number of nodes in the scenario tree; S, number of scenarios; n, number of variables; m, number of constraints nz, number of nonzero elements in the constraint matrix; $dens$, constraint matrix density in $\%$. We can observe the high dimensions of the instances: up to 7 millions of variables and 6 millions of constraints for SMALL networks, up to 78 millions of variables and 71 millions of constraints for MEDIUM networks, and up to 393 millions of variables and 357 millions of constraints for LARGE networks.

7.3 Computational results

Tables 4, 5 and 6 present for the SMALL, MEDIUM and LARGE networks the main computational results for the three LP problems SLP, EV and ELP, respectively. The headings are as follows: Z_{SLP}, Z_{EV} and Z_{ELP}, solution values of the original SLP problem and the problems EV and ELP, respectively. Note: Z_{EV} and Z_{ELP} are upper bounds of Z_{SLP}, such that $Z_{EV} \geq Z_{ELP} \geq Z_{SLP}$ (notice that SLP is a maximization problem); t_{SLP}, t_{EV} and t_{ELP}, computing time (secs.) required for obtaining Z_{SLP}, Z_{EV} and Z_{ELP}, respectively; Z_{EEV} and Z_{EELP}, solution values of the SLP feasible solutions obtained by using the EEV and EELP procedures, respectively, described in Section 6 (notice that Z_{EEV} and Z_{EELP} are lower bounds of Z_{SLP}); and t_{EEV} and t_{EELP}, computing time (secs.) required for obtaining Z_{EEV} and Z_{EELP}, respectively. Some gaps are reported as the relative differences in $\%$ of several solution values, such as

$$\begin{aligned}
GAP^1 &= \left(Z_{EV} - Z_{SLP}\right)/Z_{SLP}\% \\
GAP^2 &= \left(Z_{SLP} - Z_{EEV}\right)/Z_{SLP}\% \\
GAP^3 &= \left(Z_{EV} - Z_{EEV}\right)/Z_{EV}\% \\
GAP^4 &= \left(Z_{ELP} - Z_{SLP}\right)/Z_{SLP}\% \\
GAP^5 &= \left(Z_{SLP} - Z_{EELP}\right)/Z_{SLP}\% \\
GAP^6 &= \left(Z_{ELP} - Z_{EELP}\right)/Z_{ELP}\%.
\end{aligned}$$

Finally, $mean$ gives the average values of the gaps and the computing times in the instances considered in each table.

Our first observation is that the computing time for solving problem SLP (i.e., solving the original problem by plain use of CPLEX) generally is high for non trivial instances and, even the original problem has not been solved for instances c18, c32, c35, c37, c38, c40, c52, c54, c55, c57, c58, c59 and c60 (i.e., a 22% of our testbed), due to reach the allowed computing time (2 hours in our experimentation). Additionally, we can observe that the computing time for obtaining the upper bounds Z_{EV} and Z_{ELP} (i.e., the solution values of the problems EV and ELP, respectively) is very small, on average, even it does not reach a second. On the other side, the computing time requirements for obtaining the SLP feasible solutions whose values are Z_{EEV} and Z_{EELP} do not exceed 114 and 63 secs. for the SMALL network, respectively, 642 and 1573 secs. for the MEDIUM network, respectively, and 5210 and 7437 secs. for the LARGE network, respectively. Although the computing time required for obtaining the solution value Z_{EELP} is a bit higher than the time for obtaining Z_{EEV}, the optimality gap of the former solution value is smaller than the optimality gap of the latter one. Thus for example, in Table 5 the mean of the EELP optimality gap is $GAP^5 = 0.13\%$ while the mean of the EEV optimality gap is $GAP^2 = 1.48\%$. In any case, the gap is very small in both solution values. Finally, it is interesting to consider the quasi-optimality of the solutions values obtained by both approaches in the biggest instance (i.e., c58), being $GAP^6 = 0.48\%$ for the ELP-based approach and $GAP^3 = 3.29\%$ for the EV-based approach, being a representative comparison of the $mean$ value of the three networks considered in our experiment, see last line of Tables 4, 5 and 6.

Regarding the computational complexity notice that the number of variables and constraints in the EV, ELP and SLP problems has order $T,\ L(T-1)$ and L^{T-1}, respectively (assuming a number of possible events per stage constant and equal to L). On the other hand, the number of subproblems EV (ELP) to be solved in the EEV (EELP) approach has order L^{T-1}.

8 Conclusions

In this paper we have considered the multistage Stochastic Linear Programming (SLP) problem with a stochastic right-hand side (rhs), which is assumed to be discrete and stagewise independent. We have introduced the multistage Event Linear Programming (ELP) problem whose objective is to approximate the SLP problem. The main difference is that the former considers all the possible successions of events (scenario tree), whereas the ELP problem is bounded to only consider the events (event spike).

Roughly speaking, we have proved that the ELP problem can be derived from the SLP problem by constraint aggregation. This aggregation is obtained by applying the conditional expectation operator to the SLP constraints. In a similar way, we have proved that the Expected Value (EV) problem can be derived from the ELP problem also by constraint aggregation. In this case, the aggregation is obtained by applying the plain expectation operator to the ELP constraints. The main consequence of these aggregations is that the EV and ELP solution values are lower bounds of the SLP solution value (for minimization problems). We have also shown that the ELP bound is tighter than the EV bound.

From a practical point of view, we have assessed the validity of the ELP approach by solving large scale instances of the network revenue management problem for the flight tickets selling. We have used 60 instances whose sizes for the SLP problem range form one thousand of variables and constraints to over 393 millions of variables and 357 millions of constraints. In our computational experience the ELP approach has obtained SLP solutions whose objective values are almost optimal and significantly better than the ones obtained by the EV approach. Regarding the numerical tractability we have observed that even for large scale SLP instances the ELP heuristic is a tractable approximation, where the plain use of CPLEX very frequently fails. All in all, the main conclusion is that regarding the capacity to deal with uncertainty and the numerical tractability, the ELP approach lies between the SLP and the EV approaches.

Finally, we would like to point out some limitations of the results here presented. The first limitation, is that they only consider randomness on the right-hand side, which is assumed to be discrete and stagewise independent. The second limitation is that, although, in the computational experiment we have observed that the EEV value is dominated by the EELP value, we have not proved it theoretically. The third limitation is that this stochastic model is risk neutral. That is, it is based on the expected cost and it does not incorporate any risk averse measure as for example the conditional value at risk, or the mixture of first and second order stochastic dominance constraints. As a matter of future research, the authors are planning to improve the ELP approach regarding these aspects.

Acknowledgments: We thank the reviewers for their suggestions and comments, which helped to improve this paper. We are grateful to Jean-Philippe Vial and Alain Haurie for their comments and support at Logilab, University of Geneva, Switzerland. We also thank the support of the grant S2009/esp-1594 (Riesgos CM) from Comunidad de Madrid (Spain) and the grants MTM2012-36163-C06-06 and MTM2012-31514 from the Spanish Ministry of Economy and Competitiveness.

Table 1: Small network. Problem dimensions

					SLP				EV				ELP			
case	T	L	N	S	n	m	nz	$dens$	n	m	nz	$dens$	n	m	nz	$dens$
c1	4	2	16	8	1368	1120	2872	0.1874	432	298	736	0.5717	756	622	1564	0.3326
c2	4	3	41	27	3384	2970	7632	0.0759	432	318	736	0.5358	972	838	2176	0.2671
c3	4	4	86	64	6912	6280	16096	0.0371	432	328	736	0.5194	1188	1054	2788	0.2227
c4	5	2	32	16	2808	2312	5960	0.0918	540	370	916	0.4585	972	802	2032	0.2607
c5	5	3	122	81	10188	8982	23148	0.0253	540	400	916	0.4241	1260	1090	2860	0.2082
c6	5	4	342	256	27648	25192	64672	0.0093	540	410	916	0.4137	1548	1378	3688	0.1729
c7	6	2	64	32	5688	4696	12136	0.0454	648	442	1096	0.3827	1188	982	2500	0.2143
c8	6	3	365	243	30600	27018	69696	0.0084	648	482	1096	0.3509	1548	1342	3544	0.1706
c9	6	4	1366	1024	110592	100840	258976	0.0023	648	492	1096	0.3438	1908	1702	4588	0.1413
c10	7	2	128	64	11448	9464	24488	0.0226	756	514	1276	0.3284	1404	1162	2968	0.1819
c11	7	3	1094	729	91836	81126	209340	0.0028	756	564	1276	0.2993	1836	1594	4228	0.1445
c12	7	4	5462	4096	442368	403432	1036192	0.0006	756	574	1276	0.2940	2268	2026	5488	0.1194
c13	8	2	256	128	22968	19000	49192	0.0113	864	586	1456	0.2876	1620	1342	3436	0.1580
c14	8	3	3281	2187	275544	243450	628272	0.0009	864	646	1456	0.2609	2124	1846	4912	0.1253
c15	8	4	21846	16384	1769472	1613800	4145056	0.0001	864	656	1456	0.2569	2628	2350	6388	0.1034
c16	9	2	512	256	46008	38072	98600	0.0056	972	658	1636	0.2558	1836	1522	3904	0.1397
c17	9	3	9842	6561	826668	730422	1885068	0.0003	972	728	1636	0.2312	2412	2098	5596	0.1106
c18	9	4	87382	65536	7077888	6455272	16580512	0.0000	972	738	1636	0.2281	2988	2674	7288	0.0912
c19	10	2	1024	512	92088	76216	197416	0.0028	1080	730	1816	0.2303	2052	1702	4372	0.1252
c20	10	3	29525	19683	2480040	2191338	5655456	0.0001	1080	810	1816	0.2076	2700	2350	6280	0.0990

Table 2: Medium network. Problem dimensions

					SLP				EV				ELP			
case	T	L	N	S	n	m	nz	$dens$	n	m	nz	$dens$	n	m	nz	$dens$
c21	4	2	16	8	15200	12400	32000	0.0170	4800	3300	8200	0.0518	8400	6900	17400	0.0300
c22	4	3	41	27	37600	32900	85000	0.0069	4800	3500	8200	0.0488	10800	9300	24200	0.0241
c23	4	4	86	64	76800	69600	179200	0.0034	4800	3600	8200	0.0475	13200	11700	31000	0.0201
c24	5	2	32	16	31200	25600	66400	0.0083	6000	4100	10200	0.0415	10800	8900	22600	0.0235
c25	5	3	122	81	113200	99500	257800	0.0023	6000	4400	10200	0.0386	14000	12100	31800	0.0188
c26	5	4	342	256	307200	279200	720000	0.0008	6000	4500	10200	0.0378	17200	15300	41000	0.0156
c27	6	2	64	32	63200	52000	135200	0.0041	7200	4900	12200	0.0346	13200	10900	27800	0.0193
c28	6	3	365	243	340000	299300	776200	0.0008	7200	5300	12200	0.0320	17200	14900	39400	0.0154
c29	6	4	1366	1024	1228800	1117600	2883200	0.0002	7200	5400	12200	0.0314	21200	18900	51000	0.0127
c30	7	2	128	64	127200	104800	272800	0.0020	8400	5700	14200	0.0297	15600	12900	33000	0.0164
c31	7	3	1094	729	1020400	898700	2331400	0.0003	8400	6200	14200	0.0273	20400	17700	47000	0.0130
c32	7	4	5462	4096	4915200	4471200	11536000	0.0001	8400	6300	14200	0.0268	25200	22500	61000	0.0108
c33	8	2	256	128	255200	210400	548000	0.0010	9600	6500	16200	0.0260	18000	14900	38200	0.0142
c34	8	3	3281	2187	3061600	2696900	6997000	0.0001	9600	7100	16200	0.0238	23600	20500	54600	0.0113
c35	8	4	21846	16384	19660800	17885600	46147200	0.00001	9600	7200	16200	0.0234	29200	26100	71000	0.0093
c36	9	2	512	256	511200	421600	1098400	0.0005	10800	7300	18200	0.0231	20400	16900	43400	0.0126
c37	9	3	9842	6561	9185200	8091500	20993800	0.00003	10800	8000	18200	0.0211	26800	23300	62200	0.0100
c38	9	4	87382	65536	78643200	71543200	184592000	0.000003	10800	8100	18200	0.0208	33200	29700	81000	0.0082
c29	10	2	1024	512	1023200	844000	2199200	0.0003	12000	8100	20200	0.0208	22800	18900	48600	0.0113
c40	10	3	29525	19683	27556000	24275300	62984200	0.000009	12000	8900	20200	0.0189	30000	26100	69800	0.0089

Table 3: Large network. Problem dimensions

					SLP				EV				ELP			
$case$	T	L	N	S	n	m	nz	$dens$	n	m	nz	$dens$	n	m	nz	$dens$
c41	4	2	16	8	76000	62000	160000	0.0034	24000	16500	41000	0.0104	42000	34500	87000	0.0060
c42	4	3	41	27	188000	164500	425000	0.0014	24000	17500	41000	0.0098	54000	46500	121000	0.0048
c43	4	4	86	64	384000	348000	896000	0.0007	24000	18000	41000	0.0095	66000	58500	155000	0.0040
c44	5	2	32	16	156000	128000	332000	0.0017	30000	20500	51000	0.0083	54000	44500	113000	0.0047
c45	5	3	122	81	566000	497500	1289000	0.0005	30000	22000	51000	0.0077	70000	60500	159000	0.0038
c46	5	4	342	256	1536000	1396000	3600000	0.0002	30000	22500	51000	0.0076	86000	76500	205000	0.0031
c47	6	2	64	32	316000	260000	676000	0.0008	36000	24500	61000	0.0069	66000	54500	139000	0.0039
c48	6	3	365	243	1700000	1496500	3881000	0.0002	36000	26500	61000	0.0064	86000	74500	197000	0.0031
c49	6	4	1366	1024	6144000	5588000	14416000	0.0000	36000	27000	61000	0.0063	106000	94500	255000	0.0025
c50	7	2	128	64	636000	524000	1364000	0.0004	42000	28500	71000	0.0059	78000	64500	165000	0.0033
c51	7	3	1094	729	5102000	4493500	11657000	0.0001	42000	31000	71000	0.0055	102000	88500	235000	0.0026
c52	7	4	5462	4096	24576000	22356000	57680000	0.00001	42000	31500	71000	0.0054	126000	112500	305000	0.0022
c53	8	2	256	128	1276000	1052000	2740000	0.0002	48000	32500	81000	0.0052	90000	74500	191000	0.0028
c54	8	3	3281	2187	15308000	13484500	34985000	0.0000	48000	35500	81000	0.0048	118000	102500	273000	0.0023
c55	8	4	21846	16384	98304000	89428000	230736000	0.0000	48000	36000	81000	0.0047	146000	130500	355000	0.0019
c56	9	2	512	256	2556000	2108000	5492000	0.0001	54000	36500	91000	0.0046	102000	84500	217000	0.0025
c57	9	3	9842	6561	45926000	40457500	104969000	0.00001	54000	40000	91000	0.0042	134000	116500	311000	0.0020
c58	9	4	87382	65536	393216000	357716000	922960000	0.0000007	54000	40500	91000	0.0042	166000	148500	405000	0.0016
c59	10	2	1024	512	5116000	4220000	10996000	0.00005	60000	40500	101000	0.0042	114000	94500	243000	0,0023
c60	10	3	29525	19683	137780000	121376500	314921000	0.000002	60000	44500	101000	0.0038	150000	130500	349000	0.0018

Table 4: Small network. Solution values, gaps and computing time of SLP, EV and ELP

	SLP		EV							ELP						
case	Z_{SLP}	t_{SLP}	Z_{EV}	t_{EV}	GAP^1	Z_{EEV}	t_{EEV}	GAP^2	GAP^3	Z_{ELP}	t_{ELP}	GAP^4	Z_{EELP}	t_{EELP}	GAP^5	GAP^6
c1	298397	0.09	305046	0.02	2.23	294755	0.03	1.22	3.49	298649	0.04	0.08	297902	0.06	0.17	0.25
c2	297742	0.08	306271	0.03	2.86	293360	0.09	1.47	4.4	297994	0.04	0.08	297742	0.1	0.00	0.08
c3	295019	0.14	304640	0.03	3.26	292262	0.07	0.93	4.24	295139	0.04	0.04	294985	0.11	0.01	0.05
c4	463960	0.07	473197	0.03	1.99	455842	0.15	1.75	3.81	467008	0.04	0.66	462997	0.12	0.21	0.86
c5	457874	0.26	470839	0.02	2.83	450424	0.2	1.63	4.53	459281	0.05	0.31	456568	0.23	0.29	0.59
c6	447213	0.71	462278	0.02	3.37	440700	0.65	1.46	4.9	449534	0.04	0.52	446962	0.67	0.06	0.57
c7	425392	0.11	434504	0.02	2.14	420093	0.21	1.25	3.43	427237	0.05	0.43	424376	0.16	0.24	0.67
c8	414977	0.86	426292	0.02	2.73	409571	0.69	1.3	4.08	415704	0.05	0.18	413801	0.58	0.28	0.46
c9	414092	6.25	428134	0.02	3.39	407476	2.54	1.6	5.07	415402	0.05	0.32	413654	1.53	0.11	0.42
c10	516762	0.22	522449	0.02	1.10	511295	0.34	1.06	2.18	520202	0.04	0.67	516145	0.35	0.12	0.78
c11	506558	5.73	516355	0.03	1.93	499782	2.24	1.34	3.32	508121	0.06	0.31	505744	1.57	0.16	0.47
c12	507047	127.21	517379	0.03	2.04	499073	11.25	1.57	3.67	510288	0.05	0.64	506808	6.56	0.05	0.68
c13	483990	0.66	491188	0.03	1.49	477511	0.68	1.34	2.86	486153	0.04	0.45	483510	0.55	0.10	0.54
c14	469212	39.97	480009	0.03	2.30	462587	6.40	1.41	3.77	470299	0.05	0.23	468213	2.91	0.21	0.44
c15	470862	5646.57	480804	0.02	2.11	463480	44.76	1.57	3.74	473014	0.06	0.46	470471	23.42	0.08	0.54
c16	555201	1.10	557454	0.03	0.41	546846	0.65	1.5	1.94	555651	0.06	0.08	555061	0.64	0.03	0.11
c17	557990	412.72	562390	0.03	0.79	549226	15.62	1.57	2.4	558684	0.06	0.12	557497	11.67	0.09	0.21
c18	–	7200	560025	0.03	–	547571	113.54	–	2.27	558248	0.06	–	556965	62.77	–	0.23
c19	539579	3.70	543835	0.02	0.79	532709	1.96	1.27	2.09	540477	0.05	0.17	539232	1.12	0.06	0.23
c20	543657	6317.70	548518	0.02	0.89	538080	53.15	1.03	1.94	544706	0.06	0.19	543151	34.59	0.09	0.29
mean		988		0.03	2.03		12.76	1.38	3.41		0.05	0.31		7.49	0.12	0.42

Note: The symbol – means that the allowed computing time (7200 secs.) has been reached without obtaining an optimal solution.

Table 5: Medium network. Solution values, gaps and computing time of SLP, EV and ELP

	SLP		EV							ELP						
case	Z_{SLP}	t_{SLP}	Z_{EV}	t_{EV}	GAP^1	Z_{EEV}	t_{EEV}	GAP^2	GAP^3	Z_{ELP}	t_{ELP}	GAP^4	Z_{EELP}	t_{EELP}	GAP^5	GAP^6
c21	3472340	1	3551301	0.06	2.27	3410262	0.16	1.79	3.97	3472509	0.11	0.00	3467708	0.23	0.13	0.14
c22	3453572	1	3559792	0.08	3.08	3414416	0.35	1.13	4.08	3453835	0.13	0.01	3448635	0.51	0.14	0.15
c23	3430490	2	3542774	0.07	3.27	3384713	0.48	1.33	4.46	3430920	0.16	0.01	3428499	0.98	0.06	0.07
c24	5049137	1	5163048	0.08	2.26	4968724	0.31	1.59	3.76	5066927	0.11	0.35	5041950	0.49	0.14	0.49
c25	5025903	5	5162862	0.07	2.73	4956188	0.63	1.39	4.00	5038591	0.14	0.25	5015697	1.58	0.20	0.45
c26	4967606	39	5127061	0.07	3.21	4893424	1.80	1.49	4.56	4985313	0.18	0.36	4963942	4.14	0.07	0.43
c27	4591172	2	4694998	0.08	2.26	4518853	0.87	1.58	3.75	4610553	0.13	0.42	4582732	1.17	0.18	0.60
c28	4556401	33	4695780	0.09	3.06	4484627	2.08	1.58	4.50	4568933	0.17	0.28	4548538	5.46	0.17	0.45
c29	4553812	401	4695066	0.08	3.10	4486527	6.86	1.48	4.44	4572962	0.26	0.42	4548207	18.44	0.12	0.54
c30	5816031	5	5926436	0.09	1.90	5732435	1.01	1.44	3.27	5842266	0.14	0.45	5810633	2.01	0.09	0.54
c31	5814053	413	5950584	0.09	2.35	5731749	7.03	1.42	3.68	5836210	0.22	0.38	5804662	19.57	0.16	0.54
c32	–	7200	5924682	0.09	–	5697159	31.42	–	3.84	5809691	0.29	–	5776579	84.38	–	0.57
c33	5430836	16	5548936	0.10	2.17	5348683	3.36	1.51	3.61	5459660	0.16	0.53	5425680	4.82	0.09	0.62
c34	5402733	3323	5541490	0.10	2.57	5328377	23.78	1.38	3.85	5421152	0.26	0.34	5390441	61.39	0.23	0.57
c35	–	7200	5556352	0.10	–	5322904	145.02	–	4.20	5422905	0.33	–	5390816	377.44	–	0.59
c36	6225830	51	6312576	0.10	1.39	6131050	6.32	1.52	2.88	6253568	0.17	0.45	6220926	10.19	0.08	0.52
c37	–	7200	6291115	0.10	–	6077781	85.26	–	3.39	6207474	0.29	–	6174479	219.57	–	0.53
c38	–	7200	6307562	0.10	–	6086270	641.20	–	3.51	6219477	0.37	–	6188692	1572.38	–	0.49
c49	6234951	158	6317934	0.11	1.33	6137151	14.03	1.57	2.86	6262235	0.21	0.44	6230973	22.97	0.06	0.50
c40	–	7200	6295263	0.11	–	6080922	260.89	–	3.40	6208549	0.33	–	6176582	706.03	–	0.51
mean		2023		0.09	2.46		61.64	1.48	3.80		0.21	0.31		155.69	0.13	0.47

Note: The symbol – means that the allowed computing time (7200 secs.) has been reached without obtaining an optimal solution.

Table 6: Large network. Solution values, gaps and computing time of SLP, EV and ELP

	SLP		EV							ELP						
case	Z_{SLP}	t_{SLP}	Z_{EV}	t_{EV}	GAP^1	Z_{EEV}	t_{EEV}	GAP^2	GAP^3	Z_{ELP}	t_{ELP}	GAP^4	Z_{EELP}	t_{EELP}	GAP^5	GAP^6
c41.	17494863	15	17843110	0.32	1.99	17218508	0.82	1.58	3.50	17496618	0.35	0.01	17461781	1.14	0.19	0.2
c42	17248812	22	17773243	0.31	3.04	17044274	1.7	1.19	4.10	17250177	0.67	0.01	17220992	2.55	0.16	0.17
c43	17230686	44	17783665	0.31	3.21	16986650	3.62	1.42	4.48	17232538	0.73	0.01	17210077	5.88	0.12	0.13
c44	25222490	19	25774368	0.37	2.19	24861783	1.65	1.43	3.54	25308628	0.47	0.34	25187420	1.9	0.14	0.48
c45	25047326	75	25743179	0.37	2.78	24699115	5.44	1.39	4.06	25114873	0.7	0.27	25007005	8.97	0.16	0.43
c46	25014633	533	25779709	0.39	3.06	24660763	13.38	1.41	4.34	25106633	0.88	0.37	24996473	23.6	0.07	0.44
c47	23086497	28	23601801	0.38	2.23	22724655	4.29	1.57	3.72	23173802	0.58	0.38	23040551	5.33	0.20	0.58
c48	22898282	322	23555506	0.39	2.87	22567523	21.96	1.44	4.19	22966517	1	0.30	22850346	37.39	0.21	0.51
c49	22873137	4123	23587413	0.39	3.12	22535803	75.63	1.47	4.46	22965296	1.32	0.40	22841067	161.86	0.14	0.54
c50	29179564	70	29729277	0.42	1.88	28788185	9.38	1.34	3.17	29314894	0.66	0.46	29151588	11.26	0.10	0.56
c51	28959292	2896	29630700	0.42	2.32	28552069	70.99	1.41	3.64	29066938	1.2	0.37	28910766	153.23	0.17	0.54
c52	–	7200	29756274	0.42	–	28623996	332.87	–	3.81	29169591	1.52	–	29018903	423.89	–	0.52
c53	27291635	191	27868780	0.46	2.11	26900986	21.49	1.43	3.47	27431715	0.81	0.51	27261469	26.33	0.11	0.62
c54	–	7200	27807372	0.47	–	26737727	255.94	–	3.85	27212077	1.46	–	27061944	300.39	–	0.55
c55	–	7200	27837640	0.47	–	26705838	1449.96	–	4.07	27225733	1.93	–	27060239	2129.44	–	0.61
c56	31086619	768	31504473	0.48	1.34	30646873	44.47	1.41	2.72	31203773	0.85	0.38	31065100	53.64	0.07	0.44
c57	–	7200	31462806	0.56	–	30517895	779.74	–	3.00	31068833	1.74	–	30925113	1258.97	–	0.46
c58	–	7200	31517734	0.64	–	30482106	5209.12	–	3.29	31076927	2.08	–	30927563	7436 .34	–	0.48
c59	–	7200	31542104	0.62	–	30657955	80.56	–	2.80	31220519	1.02	–	30927563	120.32	–	0.44
c60	–	7200	31465398	0.64	–	30522381	1998.23	–	3.00	31057137	2.08	–	30909264	4262.74	–	0.48
mean		2975		0.44	2.47		519.06	1.42	3.66		1.10	0.29		821.24	0.14	0.46

Note: The symbol – means that the allowed computing time (7200 secs.) has been reached without obtaining an optimal solution.

References

[1] D. Bertsimas and I. Popescu. Revenue management in a dynamic network environment. *Transportation Science*, 37(3):257–277, 2003.

[2] J. R. Birge. Aggregation bounds in stochastic linear programming. *Mathematical Programming*, 31(1):25–41, 1985.

[3] J. R. Birge. Stochastic programming: Computation and applications. *INFORMS Journal on Computing*, 9(2):111–133, 1997.

[4] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, 2nd edition, 2011.

[5] M. S. Casey and S. Sen. The scenario generation algorithm for multistage stochastic linear programming. *Mathematics of Operations Research*, 30:615–631, 2005.

[6] M. Colombo, J. Gondzio, and A. Grothey. A warm-start approach for large-scale stochastic linear programs. *Mathematical Programming*, 127(2):371–397, 2011.

[7] M. Davidson. Primal-dual constraint aggregation with application to stochastic programming. *Annals of Operations Research*, 99:41–58, 2000.

[8] M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. Addison Wesley, 2002.

[9] J. Dupacova, G. Consigli, and S. W. Wallace. Scenarios for multistage stochastic programs. *Annals of Operations Research*, 100:25–53, 2000.

[10] L. F. Escudero, A. Garin, M. Merino, and G. Perez. The value of the sochastic solution in multistage problems. *TOP (Journal of the Spanish Statistics and Operations Research Society)*, 15:48–64, 2007.

[11] L. F. Escudero, M. A. Garin, M. Merino, and G. Perez. An algorithmic framework for solving large-scale multistage stochastic mixed 0–1 problems with nonsymmetric scenario trees. *Computers & Operations Research*, 39(5):1133–1144, 2012.

[12] L. F. Escudero, J. F. Monge, D. Romero-Morales, and J. Wang. Expected future value decomposition based bid price generation for large-scale network revenue management. *Transportation Science*, 47:181–197, 2013.

[13] L.F. Escudero, J.L. de la Fuente, C. Garcia, and F.J. Prieto. A parallel computation approach for solving multistage stochastic network problems. *Annals of Operations Research*, 90:131–160, 1999.

[14] E. Fragniere, J. Gondzio, and J-Ph. Vial. Building and solving large-scale stochastic programs on an affordable distributed computing system. *Annals of Operations Research*, 99:167–187, 2000.

[15] H. Heitsch and W. Romisch. Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118(2):371–406, 2009.

[16] IBM ILOG CPLEX Optimizer,
www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[17] P. Kall and J. Mayer. *Stochastic Linear Programming*. Kluwer academic publishers, 2005.

[18] S. Karmarkar, D. Goutam, and B. Tathagata. Revenue impacts of demand unconstraining and accounting for dependency. *Journal of Revenue and Pricing Management*, 10(4):367–381, 2011.

[19] C. Kuchler and S. Vigerske. Decomposition of multistage stochastic programs with recombining scenario trees. Technical Report 9, Stochastic Programming E-Print Series (SPEPS), 2007.

[20] D. Kuhn. Aggregation and discretization in multistage stochastic programming. *Mathematical Programming*, 113(1):61–94, 2008.

[21] J. Linderoth and S. Wright. Decomposition algorithms for stochastic programming on a computational grid. *Computational Optimization and Applications*, 24(2):207–250, 2003.

[22] F. Maggioni, E. Allevi, and M. Bertocchi. The value of information in multistage linear stochastic programming. *Journal of Optimization Theory and Applications*, (accepted) 2013.

[23] A. Moller, W. Romisch, and K. Weber. A new approach to OD revenue management based on scenario trees. *Journal of Revenue and Pricing Management*, 3(3):265–276, 2004.

[24] J. M. Mulvey, R. J. Vanderbei, and S.A. Zenios. Robust optimization of large-scale systems. *Operations Research*, 43(2):264–281, 1995.

[25] T. Pennanen. Epi-convergent discretizations of multistage stochastic programs via integration quadratures. *Mathematical Programming, Ser. B*, 116:461–479, 2009.

[26] A. B. Philpott and V. L. de Matos. Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. *European Journal of Operational Research*, 218(2):470–483, 2012.

[27] W. B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. Wiley-Interscience, 2007.

[28] A. Prekopa. *Stochastic Programming*. Springer, 1995.

[29] S. Sen and Z. Zhou. Multistage stochastic decomposition: a bridge between stochastic programming and approximate dynamic programming. *SIAM Journal on Optimization*, 24(1):127–153, 2014.

[30] A. Shapiro. Inference of statistical bounds for multistage stochastic programming problems. *Mathematical Methods of Operations Research*, 58(1):57–68, 2003.

[31] A. Shapiro. On complexity of multistage stochastic programs. *Operations Research Letters*, 34(1):1–8, 2006.

[32] A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory*, volume 9. Society for Industrial Mathematics, 2009.

[33] J. Thenie and J.-Ph. Vial. Step decision rules for multistage stochastic programming: a heuristic approach. *Automatica*, 44(6):1569–1584, 2008.

[34] H. Topaloglu. A tighter variant of Jensen's lower bound for stochastic programs and separable approximations to recourse functions. *European Journal of Operational Research*, 199(2):315–322, 2009.

[35] S. E. Wright. Primal-dual aggregation and disaggregation for stochastic linear programs. *Mathematics of Operations Research*, 19(4):893–908, 1994.