On a Stochastic Bilevel Programming Problem with Knapsack Constraints

Stefanie Kosuch[*] Pierre Le Bodic[*] Janny Leung[◇] Abdel Lisser[*]

[*] *Laboratoire de Recherche en Informatique (LRI), Université Paris Sud - Paris XI 91405 Orsay Cedex - France*

[◇] *Department of Systems Engineering & Engineering Management, Chinese University of Hong Kong Shatin, NT - Hong Kong SAR - The People's Republic of China*

Abstract

In this paper we propose a mixed integer bilevel problem having a probabilistic knapsack constraint in the first level. The problem formulation is mainly motivated by practical pricing and service provision problems as it can be interpreted as a model for the interaction between a service provider and clients. We assume the probability space to be discrete which allows us to reformulate the problem as an equivalent deterministic bilevel problem. By the mean of a reformulation as linear bilevel problem, we obtain a quadratic optimization problem, the so called Global Linear Complementarity Problem. Based on this quadratic problem, we propose a procedure to compute upper bounds on the initial problem by using a Lagrangian relaxation and an iterative linear minmax scheme. Finally, numerical experiments confirm that the scheme practically converges.

Keywords: *Bilevel, Stochastic, Optimization, Networks, Pricing*

1 Introduction

In network markets, service providers are increasingly presented with optimization problems including not only network capacity or routing constraints but also market interactions with competitors [24, 3, 6, 17]. Pricing and capacity decisions are hardly ever separated: on the one hand, the optimal utilisation of the capacity of a given network depends on the demand. On the other hand, one cannot properly adapt its prices to the market if its network is fixed, since the demand will most likely vary according to the prices. It is therefore natural to formulate the problem of pricing one's services under the network constraints at hand as a bilevel optimization problem [12, 9, 7]: while the *leader* (e.g. a

service provider) maximizes its profit, the *follower* (e.g. customers) minimizes the cost of the services by choosing among a set of competitors. The follower's problem is hence a constraint of the leader's problem.
In the deterministic bilevel literature, pricing in networks problems are usually modeled with bilinear objective functions (see e.g. [22, 10, 5]). In our study of this problem we however assume that the cost and pricing functions as well as the network and capacity constraints are linear (cf. subsection 1.1 for a network application).
In the networks considered, we assume that the demand is uncertain (e.g. internet traffic). The leader can thus optimize its profit by replacing the deterministic demand-satisfying constraint by a probability constraint. The complete problem can be stated as a Stochastic Linear Bilevel Problem:

$$
\begin{aligned}
\text{(SLBP)} \quad & \max_x \quad c_1^t x + d_1^t y && \text{(1a)}\\
& \text{s.t. } A^1 x + B^1 y \leq b^1, && \text{(1b)}\\
& \qquad \mathbb{P}\{w^t(\omega) x \leq s(\omega)\} \geq (1-\alpha), && \text{(1c)}\\
& \qquad 0 \leq x \leq \mathbb{1}_{n_x}, && \text{(1d)}\\
& \qquad y \in \arg\max_y c_2^t x + d_2^t y, && \text{(1e)}\\
& \qquad \text{s.t. } A^2 x + B^2 y \leq b^2, && \text{(1f)}\\
& \qquad\qquad y \geq 0. && \text{(1g)}
\end{aligned}
$$

where x (resp. y) is the vector of n_x (resp. n_y) decision variables of the leader (resp. follower), $c_1, c_2, \in \mathbb{R}^{n_x}$, $d_1, d_2 \in \mathbb{R}^{n_y}$, $A^1 \in \mathbb{R}^{m_1 \times n_x}$, $B^1 \in \mathbb{R}^{m_1 \times n_y}$, $b^1 \in \mathbb{R}^{m_1}$, $A^2 \in \mathbb{R}^{m_2 \times n_x}$, $B^2 \in \mathbb{R}^{m_2 \times n_y}$, $b^2 \in \mathbb{R}^{m_2}$ and $0 < \alpha \leq 1$. $\mathbb{1}_n$ denotes the vector of ones of dimension n. The components of $w(\omega) \in \mathbb{R}_+^{n_x}$ and the right hand side $s(\omega) \in \mathbb{R}$ are random variables that all depend on the realization ω of the probability space Ω.
Both the objective function of the leader (1a) and the follower (1e) depend on the leader's strategy x *and* the follower's strategy y. The probabilistic knapsack constraint (1c) ensures that customers will be served with a α risk. Note that this model as well as the results presented in this paper can easily be extended to multiple stochastic knapsack constraints. Constraints (1f) are relative to the customers demand.
Throughout this paper, we will study the general SLBP case. We show in the following application how this work can be used to tackle pricing problems in networks. In particular, we linearize the bilinear objective functions of an existing application model. The modeling method we provide requires an additional reasonable hypothesis, but might be used to linearize other such problems.

1.1 A pricing in networks application

We will now illustrate the general model (1) with a network application, motivated by the deterministic application presented in [5]. Let us describe without loss of generality a context-simplified version of the application considered in

[5]. An operator rents capacities on an oriented network with capacitated arcs to a set of customers. A client can split its demand between the operator and the competition. Their demand is formulated as a *market*, i.e. couple of origin and destination vertices on the network. From the provider's point of view, a single predetermined path is used to serve each market. For a given market, the operator and the competition have each a single offer. We suppose that the competition can serve the demand of all the clients and that its tariffs are known. The operator sets the prices and the capacity allocated to a market. The capacity of an arc is split between markets. Clients minimize the total cost to satisfy the demand, by choosing between the operator and the competition. The application we present derive directly from this real-world application, by adding stochasticity to a previously deterministic capacity constraint. Also, we make the assumption that a client cannot split its demand. This might seem like a strong hypothesis, however this is in general the case when each client has a small demand: it may be neither necessary nor financially interesting to contract multiple services. This hypothesis will make it possible to linearize the bilinear objective functions. The constraint ensuring that the capacity of markets are respected was moved from the lower-level to the upper-level problem. From the application point of view, this means that the provider cannot refuse clients. Therefore, the only possibility to adjust the number of clients is to change the prices.

Let us now formally state the application model. Let a network $N = (V, A)$ together with the following parameters be given:

- A set of source/sink pairs $\mathcal{M} \subseteq V^2$ (markets)
- A path p_m from source to sink for all $m \in \mathcal{M}$
- A maximum capacity c_a for all $a \in A$

On this network, an operator o as well as a group of competitors c allocate capacity to a set of clients $\mathcal{S}$. Each client $s \in \mathcal{S}$ is interested in exactly one market $m_s \in \mathcal{M}$. Each client has a maximum demand d_s which defines the demand vector $d \in \mathbb{R}_+^{|\mathcal{S}|}$. We assume that operator o knows the minimum price T_s^c that the competition proposes to client s. We can therefore also assume that the operator itself has an upper bound Γ_s^o on the price he fixes for client s.

We introduce a random variable $w(\omega) \in [0,1]^{|\mathcal{S}|}$ representing the fraction of the capacity actually used by the clients as this demand is assumed not to be known in advance. In order to optimally use the network we allow a risk α of an overbooking.

In this bilevel problem the leader is the operator o. He has to fix its price for each of the clients and the amount of capacity allocated to the different markets. The leader's decision variables are defined as follows:

- T_s^o: price of the operator for client s
- t_s: price that has to be paid by client s to the operator
- C_m: capacity allocated for a given market $m \in \mathcal{M}$

The follower is the group of clients: each client has to decide whether to buy from operator o (represented by setting the follower's decision variable o_s to 1) or from the competition ($o_s = 0$).
This problem can be modeled by the following stochastic linear bilevel problem:

$$\max_{T^o,t,C} \sum_{s\in\mathcal{S}} t_s \tag{2a}$$

$$\text{s.t.} \quad t_s \le d_s T^o_s \qquad \forall s \in \mathcal{S} \tag{2b}$$

$$t_s \le d_s o_s \Gamma_s \qquad \forall s \in \mathcal{S} \tag{2c}$$

$$\sum_{m\in\mathcal{M}:a\in p_m} C_m \le c_a \qquad \forall a \in A \tag{2d}$$

$$\mathbb{P}\{ \sum_{s\in\mathcal{S}:m_s=m} w_s(\omega) d_s o_s \le C_m\} \ge 1-\alpha \qquad \forall m \in \mathcal{M} \tag{2e}$$

$$o \in \arg\min_{o\in\{0,1\}^{|\mathcal{S}|}} \sum_{s\in\mathcal{S}} t_s + \sum_{s\in\mathcal{S}} T^c_s d_s (1-o_s) \tag{2f}$$

The objective function (2a) means that the operator maximizes the total price paid by clients. Constraints (2b) and (2c) are a linearization of the bilinear term $d_s T^o_s o_s$, replaced by t_s in both objective functions. Constraints (2d) splits the capacity of the arcs between markets. The stochastic constraints (2e) let the leader overbook market capacities with a given overload risk. The follower's problem (2f) reduces to choosing the least expensive solution.
Remark that in this bilevel problem the follower's decision variables appear in the probability constraint. However, all results and models presented in this paper can easily be extended to the case where the probability constraint contains any subset of the union of leader's and follower's decision variables. Also, note that even though the lower-level problem possesses binary variables, it is equivalent to consider continuous variables due to its structure. As in the original application, we adopt an optimistic tie-breaking rule.

1.2 Literature review and solution method

Stochastic and bilevel aspects of mathematical programming have been very little studied together, with the exception of the pioneering work of [23, 29, 8, 33, 3]. Most of these works are interested in network problems. For single-level stochastic problems, please refer to [16].
A very recent publication concerning stochastic bilevel problems with knapsack constraint is that of Özaltin et al. [27]. The authors solve a stochastic version of the bilevel knapsack problem studied by Dempe and Richter in [11]. In their variant of the problem the decision of the leader consists in choosing the (one-dimensional) right hand side of the knapsack constraint (i.e. the capacity of the knapsack). Depending on this value, the follower has to solve a common knapsack problem. Özaltin et al. extend this model by introducing an uncertainty in the lower-level problem. More precisely, they assume that the right hand side of the knapsack constraint in the lower-level does not only depend on the leader's

decision but also on a random variable.
Throughout, we will assume the probability space Ω to be discrete, i.e. having a finite number of realizations, called scenarii. From a practical point of view, one can imagine the service provider to lend capacity in packages rather than continuously, i.e. customers have to choose between several options of maximal needed capacity. If, in contrary, the customers are free to use the network to route their commodities, i.e. if the actual probability space is continuous, our assumption and the resulting problem reformulations might nevertheless be helpful as one could approximate the probability space by generating a finite number of representative scenarii. From a theoretical point of view, assuming a finite sample space has the advantage that the problem can be reformulated as a deterministic equivalent problem by treating the constraints for every scenario separately. In our case this leads to a Mixed Integer Linear Bilevel Problem (MILBP).
MILBPs have been studied in [32] and the authors show that in case where integer variables only appear in the first level, the problem can be equivalently reformulated as a Linear Bilevel Problem (LBP). In [15] the authors show how to further convert an LBP into a single level, bilinear program. We apply both techniques to our problem. However, our objective is not to solve the resulting quadratic problem exactly, which could for example be achieved by reformulating the problem as an MIP (see [15]) or by using other quadratic programming methods. Instead, we propose upper bounds by relaxing the bilinear constraints in the objective function. This gives rise to a bilinear min-max problem of the following form: The constraints are linear and separable, i.e. there exists a partition of the variables that also partitions the constraints into two disjoint sets. When fixing one of these sets of variables, the resulting problem is either a linear minimization, or maximization problem.
Bilinear problems of this type have been studied in the literature (see [19]). However, to the best of our knowledge all of these studies concerned pure minimization (resp. maximization) problems known as *bilinear programming problems with separate (or disjoint) constraints*. Due to the structure of these problems it is obvious why the decomposition of the initial problem is the basis of most solution methods. By simply iteratively solving the two subproblems (or the primal of one and the dual of the other) one however cannot ensure global con vergence. The most common approaches to guarantee that the algorithm converges towards a global maximum are cutting plane ([20],[30]) and branch-and-bound ([13],[2]) methods (for a survey see e.g. [1] or [14]).
As in our case one of the subproblems is a maximization, while the other is a minimization problem, we propose a minmax scheme that alternately solves the two problems. The convergence to the global optimum of the bilinear minmax problem is ensured by continuously introducing new cuts to both problems.
There has been an extensive research on minmax problems. A paper that treats a minmax problem with a similar structure to our problem is [28]: as in our case, the studied minmax problem has separable linear constraints and a quadratic objective function. The solution procedure proposed is a finite iteration method that, in each iteration, solves a quadratic subproblem. Although we cannot

guarantee that our algorithm finds the optimal solution in finitely many iterations, its advantages are the simple structure and the fact that we have to solve a sequence of linear programs, instead of a sequence of quadratic programs. Furthermore, the numerical tests indicate that the sequence converges, up to a given tolerance, after finitely many steps.
The remainder of this article is structured as follows: In section 2 we replace the probabilistic constraint (1c) by a set of equivalent linear constraints using the assumption that only finitely many scenarii have non-zero probability. In section 3, we further transform the problem into a quadratic programming problem. In section 4, we suggest and analyze a method to compute upper bounds on the initial problem by solving a relaxation using an iterative minmax scheme. We further improve this scheme in section 5 in order to obtain a convergence guarantee. Finally, we conduct numerical experiments in section 6 which illustrate the good convergence properties of our method.

2 From SLBP to the (Deterministic Equivalent) Linear Bilevel Problem (LBP)

As we consider the case where the sample space Ω is finite, ω has only a finite number of scenarii $\omega_1, ..., \omega_K$. Let us define $p^k := \mathbb{P}\{\omega = \omega_k\}$, then

$$\sum_{k=1}^{K} p_k = 1, \quad p_k \geq 0$$

For each scenario $\omega_k, k = 1, \ldots, K$ we introduce an auxiliary binary variable z_k as follows:

$$z_k = \begin{cases} 0 & \text{if the scenario is considered} \\ 1 & \text{otherwise} \end{cases}$$

We shall simplify the notations by defining for all $k = 1, \ldots, K$:

$$w_k := w\left(\omega_k\right), \ s_k := s\left(\omega_k\right), \ w^k := \left(w_1^k, ..., w_n^k\right)$$

For all $k = 1, \ldots, K$, we define M_k such that

$$M_k := \sum_{i=1}^{n_x} w_i^k - s_k$$

Thus, problem (1) can be reformulated as the following mixed integer bilevel problem:

$$
\begin{aligned}
\text{(MILBP)} \quad \max_{x,z} \quad & c_1^t x + d_1^t y \\
\text{s.t. } & A^1 x + B^1 y \le b^1, && \text{(3a)}\\
& w_k^t x \le s_k + M_k z_k, \ k = 1, \dots, K, && \text{(3b)}\\
& p^t z \le \alpha, && \text{(3c)}\\
& 0 \le x \le \mathbb{1}_{n_x}, && \text{(3d)}\\
& z \in \{0,1\}^K, && \text{(3e)}\\
& y \in \arg\max_y c_2^t x + d_2^t y, && \text{(3f)}\\
& \quad \text{s.t. } A^2 x + B^2 y \le b^2, && \text{(3g)}\\
& \quad\quad y \ge 0. && \text{(3h)}
\end{aligned}
$$

Constraints (3b) ensure that, if scenario ω_k is not covered (i.e. $z_k = 1$), then the adopted strategy x does not have to respect the knapsack constraint for this scenario. However, as per constraint (3c), the probability of occurrence of the uncovered scenarii must be below the risk α. Note that

$$w_k^t x \le s_k + M z_k, \ k = 1, \dots, K$$

with $M = \max_{k=1,\dots,K} M_k$ is a more frequent manner than (3b) to write stochastic knapsack constraints. However, allowing the M_k values to be different for constraints (3b) would yield tighter LP-relaxations. As shown in [32], we can now reformulate the mixed integer bilevel problem (3) as a linear one:

$$
\begin{aligned}
\text{(LBP)} \quad \max_{x,z} \quad & c_1^t x + d_1^t y \\
\text{s.t. } & A^1 x + B^1 y \le b^1, && \text{(4a)}\\
& w_k^t x \le s_k + M_k z_k, \ k = 1, \dots, K, && \text{(4b)}\\
& p^t z \le \alpha, && \text{(4c)}\\
& 0 \le x \le \mathbb{1}_{n_x}, && \text{(4d)}\\
& 0 \le z \le \mathbb{1}_K, && \text{(4e)}\\
& v = 0, && \text{(4f)}\\
& (y, v) \in \arg\max_{y,v} c_2^t x + d_2^t y + (\mathbb{1}_K)^t v, && \text{(4g)}\\
& \quad \text{s.t. } A^2 x + B^2 y \le b^2, && \text{(4h)}\\
& \quad\quad v \le z, && \text{(4i)}\\
& \quad\quad v \le \mathbb{1}_K - z, && \text{(4j)}\\
& \quad\quad y \ge 0. && \text{(4k)}
\end{aligned}
$$

where $\dim(v) = \dim(z) = K$. The term $(\mathbb{1}_K)^t v$ in the lower-level objective function forces v to be equal to $\min(1_K - z, z)$ (see proposition 2.2). Note that the vector $\mathbb{1}_K$ could be replaced by any vector with strictly positive components.

Definition 2.1. *We denote* $(\tilde{x}, \tilde{z}, \tilde{y})$ *(resp.* $(\tilde{x}, \tilde{z}, \tilde{y}, \tilde{v})$*) a feasible solution for problem (3) (resp. problem (4)) if all upper- and lower-level constraints are satisfied. A rational solution of problem (3) (resp. (4)) is a feasible solution such that* $\tilde{y}$ *(resp.* $(\tilde{y}, \tilde{v})$*) is optimal for the lower-level problem with parameters* $\tilde{x}$ *and* $\tilde{z}$*.*

Proposition 2.2 (see proposition 3.2. of [4])**.**

1.) *Let* (x^*, z^*, y^*, v^*) *be a rational optimal solution of LBP (4).*
Then $v^* = 0$ *and* (x^*, z^*, y^*) *is a rational optimal solution of MILBP (3).*

2.) *Let* (x^*, z^*, y^*) *be a rational optimal solution of MILBP (3).*
Then $(x^*, z^*, y^*, 0)$ *is a rational optimal solution of LBP (4).*

Proof. Proof of 1.): By constraint (4f), we have $v^* = 0$. If $(x^*, z^*, y^*, 0)$ is a rational solution, then by its optimality for the lower-level problem, we have $0 = v^* = \min(1_K - z^*, z^*)$, so $z^* \in \{0, 1\}^K$. Thus, (x^*, y^*, z^*) is feasible for problem (3). By optimality of $(x^*, y^*, z^*, v^* = 0)$, (x^*, y^*, z^*) is also optimal for problem (3).
Proof of 2.): It is easy to see that $(x^*, z^*, y^*, 0)$ is a rational solution of problem (4). From 1.) we know that *every* rational optimal solution (x^*, z^*, y^*, v^*) of problem (4) satisfies $v^* = 0$. It follows that $(x^*, z^*, y^*, 0)$ is an optimal solution of problem (4).

□

3 From LBP to the Global Linear Complementarity Problem (GLCP)

We will now continue the transformation process by reformulating LBP as a single level GLCP as described in [4]. The idea is to replace the lower-level problem by a set of constraints that contain (i) the initial constraints of the lower-level problem and (ii) the complementary slackness conditions of the lower-level problem. The latter ensures that an optimal solution of the obtained single level problem is also optimal for LBP. The decision vectors of the new problem are both the decision vectors of the upper- and lower-level problems as well as the dual variables of the latter.
Let us first state the dual of the follower's problem (4g)-(4k):

$$
\begin{aligned}
\text{(DFP)} \min_{\lambda, \mu_1, \mu_2} \quad & \lambda^t (b^2 - A^2 x) + \mu_1 z + \mu_2 (\mathbb{1}_K - z), \\
\text{s.t.} \quad & (B^2)^t \lambda \geq d_2, && \text{(5a)} \\
& \mathbb{I}_K \mu_1 + \mathbb{I}_K \mu_2 \geq \mathbb{1}_K, && \text{(5b)} \\
& \lambda, \mu_1, \mu_2 \geq 0. && \text{(5c)}
\end{aligned}
$$

where $\lambda \in \mathbb{R}^{m_2}$ (resp. $\mu_1 \in \mathbb{R}^K$, $\mu_2 \in \mathbb{R}^K$) is the dual variable associated with (4h) (resp. (4i), (4j)). We also need the corresponding complementary slackness conditions to ensure the optimality of DFP:

$$\lambda^t(b^2 - A^2x - B^2y) = 0 \qquad y^t((B^2)^t\lambda - d_2) = 0$$
$$\mu_1^t(z - v) = 0 \qquad v^t(\mathbb{I}_K\mu_1 + \mathbb{I}_K\mu_2 - \mathbb{1}_K) = 0$$
$$\mu_2^t(\mathbb{1}_K - z - v) = 0$$

We obtain the following equivalent Global Linear Complementarity Problem which is no longer a bilevel problem [4]:

$$\text{(GLCP)} \quad \max_{x,y,z,\lambda,\mu_1,\mu_2} \quad c_1^t x + d_1^t y$$
$$\text{s.t. } A^1x + B^1y \leq b^1, \tag{6a}$$
$$w_k^t x \leq s_k + M_k z_k, \; k = 1, \ldots, K, \tag{6b}$$
$$p^t z \leq \alpha, \tag{6c}$$
$$A^2x + B^2y \leq b^2, \tag{6d}$$
$$(B^2)^t\lambda \geq d_2, \tag{6e}$$
$$\mathbb{I}_K\mu_1 + \mathbb{I}_K\mu_2 \geq \mathbb{1}_K, \tag{6f}$$
$$\lambda^t(b^2 - A^2x - B^2y) = 0, \tag{6g}$$
$$\mu_1^t z = 0, \tag{6h}$$
$$\mu_2^t(\mathbb{1}_K - z) = 0, \tag{6i}$$
$$y^t((B^2)^t\lambda - d_2) = 0, \tag{6j}$$
$$0 \leq x \leq \mathbb{1}_{n_x}, 0 \leq z \leq \mathbb{1}_K \tag{6k}$$
$$y, \lambda, \mu_1, \mu_2 \geq 0. \tag{6l}$$

Note that in this formulation the decision variable v has been eliminated due to the fact that $v = 0$.
All reformulations are equivalent so far, i.e. by solving the quadratic problem (6) we get an optimal solution of the initial stochastic bilevel problem (3) (provided that the probability space is discrete). Solving a generally nonconvex problem such as (6) directly is hard. Instead, we propose a method to compute upper bounds by relaxing it into a linear minmax problem.

4 Calculating upper bounds using Lagrangian relaxation

We relax the quadratic terms (6g), (6h), (6i) and (6j) of GLCP into the objective function:

$$\begin{aligned}\mathcal{L}(x, y, z, \lambda, \mu_1, \mu_2) =& c_1^t x + d_1^t y + \lambda^t(b^2 - A^2x - B^2y)\\ &+ \mu_1^t z + \mu_2^t(\mathbb{1}_K - z) + y^t((B^2)^t\lambda - d_2)\end{aligned}$$

Then the Lagrangian relaxation of GLCP (6) becomes

$$
\begin{aligned}
\text{(LGN)} \quad & \min_{\lambda,\mu_1,\mu_2} \max_{x,y,z} \quad \mathcal{L}(x,y,z,\lambda,\mu_1,\mu_2) \\
& \text{s.t. } A^1 x + B^1 y \le b^1, && \text{(7a)}\\
& \quad w_k^t x \le s_k + M_k z_k, \ k = 1,\dots,K, && \text{(7b)}\\
& \quad p^t z \le \alpha, && \text{(7c)}\\
& \quad A^2 x + B^2 y \le b^2, && \text{(7d)}\\
& \quad (B^2)^t \lambda \ge d_2, && \text{(7e)}\\
& \quad \mathbb{I}_K \mu_1 + \mathbb{I}_K \mu_2 \ge \mathbb{1}_K, && \text{(7f)}\\
& \quad 0 \le x \le \mathbb{1}_{n_x}, 0 \le z \le \mathbb{1}_K, && \text{(7g)}\\
& \quad y, \lambda, \mu_1, \mu_2 \ge 0. && \text{(7h)}
\end{aligned}
$$

Proposition 4.1. *Let $(x^*, y^*, z^*, \lambda^*, \mu_1^*, \mu_2^*)$ be an optimal solution of the Lagrangian relaxation (7).*
Then $\mathcal{L}(x^*, y^*, z^*, \lambda^*, \mu_1^*, \mu_2^*)$ is an upper bound on the optimal solution value of GLCP (6).

Proof. Let $(\widetilde{x}, \widetilde{y}, \widetilde{z}, \widetilde{\lambda}, \widetilde{\mu}_1, \widetilde{\mu}_2)$ denote an optimal solution of GLCP (6). As

$$
\begin{aligned}
b^2 - A^2\widetilde{x} - B^2\widetilde{y} &\ge 0\\
(B^2)^t \lambda^* - d_2 &\ge 0\\
\widetilde{y}, \lambda^*, \mu_1^*, \mu_2^* &\ge 0\\
0 \le \widetilde{z} &\le 1
\end{aligned}
$$

we have

$$c_1^t \widetilde{x} + d_1^t \widetilde{y} \le \mathcal{L}(\widetilde{x}, \widetilde{y}, \widetilde{z}, \lambda^*, \mu_1^*, \mu_2^*)$$

and as $(x^*, y^*, z^*, \lambda^*, \mu_1^*, \mu_2^*)$ is optimal for problem (7) we also have

$$\mathcal{L}(\widetilde{x}, \widetilde{y}, \widetilde{z}, \lambda^*, \mu_1^*, \mu_2^*) \le \mathcal{L}(x^*, y^*, z^*, \lambda^*, \mu_1^*, \mu_2^*)$$

□

The Lagrangian relaxation (7) has the nice property that by fixing either the primal or the dual variables, we obtain a linear problem. This property gives rise to the idea to solve (7) using an iterative scheme. More precisely, in iteration $N \ge 1$ we solve the following two linear problems:

- The Lagrangian subproblem LGNs(N), maximized over the primal variables.
- Problem LGNd(N), which is mainly composed of (5) (with additional constraints).

In each iteration of the scheme, an auxiliary constraint is added to both problems in order to enforce the convergence of their optimal solution values towards the optimal solution value of the relaxation (7). Remark that the so obtained decrease (resp. increase) of the objective function value of the LGNs (resp. LGNd) is only monotonic. In section 5 this matter is further discussed and we propose a method to obtain even strict convergence at each iteration.
The iteration process stops when $\beta - \gamma < \delta$ or $(\beta - \gamma)/\beta < \epsilon$ for small $\delta > 0$ and $\epsilon > 0$:

(LGNs(N))

$$\max_{\beta,x,y,z} \quad \beta$$

$$\begin{aligned}
\text{s.t.}\ & A^1x + B^1y \leq b^1, \\
& \beta \leq \mathcal{L}(x, y, z, \lambda^q, \mu_1^q, \mu_2^q) \\
& \qquad q = 0, \dots, N-1, && \text{(8a)} \\
& w_k^t x \leq s_k + M_k z_k \\
& \qquad k = 1, \dots, K, && \text{(8b)} \\
& p^t z \leq \alpha, && \text{(8c)} \\
& A^2x + B^2y \leq b^2, && \text{(8d)} \\
& 0 \leq x \leq \mathbb{1}_{n_x}, && \text{(8e)} \\
& 0 \leq z \leq \mathbb{1}_K, && \text{(8f)} \\
& y \geq 0. && \text{(8g)}
\end{aligned}$$

(LGNd(N))

$$\min_{\gamma,\lambda,\mu_1,\mu_2} \quad \gamma$$

$$\begin{aligned}
\text{s.t.}\ & \gamma \geq \mathcal{L}(x^q, y^q, z^q, \lambda, \mu_1, \mu_2) \\
& \qquad q = 1, \dots, N, && \text{(9a)} \\
& (B^2)^t \lambda \geq d_2, && \text{(9b)} \\
& \mathbb{I}_K \mu_1 + \mathbb{I}_K \mu_2 \geq \mathbb{1}_K, && \text{(9c)} \\
& \lambda, \mu_1, \mu_2 \geq 0. && \text{(9d)}
\end{aligned}$$

where $N \geq 1$ is the iteration number, (x^q, y^q, z^q) is an optimal solution of problem LGNs(q) ($q = 1, \dots, N$), $(\lambda^q, \mu_1^q, \mu_2^q)$ is feasible for problem (7) if $q = 0$ and it is an optimal solution of problem LGNd(q) if $q \geq 1$. In the remainder of this paper we will use the following notations:

N1: $X \subseteq \mathbb{R}_+^{n_x+n_y+K}$ denotes the set of triples (x, y, z) feasible for problem (7)

N2: $\Lambda \subseteq \mathbb{R}_+^{m+2K}$ denotes the set of triples (λ, μ_1, μ_2) feasible for problem (7)

N3: (β^N, x^N, y^N, z^N) denotes an optimal solution of problem LGNs(N)

N4: $(\gamma^N, \lambda^N, \mu_1^N, \mu_2^N)$ denotes an optimal solution of problem LGNd(N)

Using the first two notations, problems (8) and (9) of the iterative minmax scheme can be stated equivalently as

$$\text{(LGNs}(N)) \qquad \max_{(x,y,z)\in X} \min_{q\in\{0,\dots,N-1\}} \qquad \mathcal{L}(x, y, z, \lambda^q, \mu_1^q, \mu_2^q)$$

$$\text{(LGNd}(N)) \qquad \min_{(\lambda,\mu_1,\mu_2)\in\Lambda} \max_{q\in\{1,\dots,N\}} \qquad \mathcal{L}(x^q, y^q, z^q, \lambda, \mu_1, \mu_2)$$

We directly get the following properties:

P1: $\beta^N = \max_{(x,y,z)\in X} \min_{q\in\{0,\ldots,N-1\}} \quad \mathcal{L}(x, y, z, \lambda^q, \mu_1^q, \mu_2^q)$

P2: $\beta^N = \min_{q\in\{0,\ldots,N-1\}} \mathcal{L}(x^N, y^N, z^N, \lambda^q, \mu_1^q, \mu_2^q)$

P3: $\exists\, j_N \in \{0 \ldots N-1\}$ s.t. $\beta^N = (x^N, y^N, z^N, \lambda^{j_N}, \mu_1^{j_N}, \mu_2^{j_N})$

P4: $\gamma^N = \min_{(\lambda,\mu_1,\mu_2)\in\Lambda} \max_{q\in\{1,\ldots,N\}} \quad \mathcal{L}(x^q, y^q, z^q, \lambda, \mu_1, \mu_2)$

P5: $\gamma^N = \max_{q\in\{1,\ldots,N\}} \mathcal{L}(x^q, y^q, z^q, \lambda^N, \mu_1^N, \mu_2^N)$

P6: $\exists\, i_N \in \{1 \ldots N\}$ s.t. $\gamma^N = \mathcal{L}(x^{i_N}, y^{i_N}, z^{i_N}, \lambda^N, \mu_1^N, \mu_2^N)$

In the following, we will keep the notations j_N and i_N introduced in **P3** and **P6**.

4.1 Proving upper and lower bounds

Problems LGNs(N) and LGNd(N) provide upper and lower bounds on the minmax problem (7), respectively:

Lemma 4.2. *Let $N \geq 1$. Then γ^N is a lower bound on the optimal solution value of the Lagrangian relaxation (7).*

Proof. As $(x^q, y^q, z^q) \in X$ for all $q = 1, \ldots, N$, we have for all $(\lambda, \mu_1, \mu_2) \in \Lambda$

$$\max_{(x,y,z)\in X} \mathcal{L}(x, y, z, \lambda, \mu_1, \mu_2) \geq \max_{q=1,\ldots,N} \mathcal{L}(x^q, y^q, z^q, \lambda, \mu_1, \mu_2)$$

It follows

$$\begin{aligned}\min_{(\lambda,\mu_1,\mu_2)\in\Lambda} \max_{(x,y,z)\in X} \mathcal{L}(x, y, z, \lambda, \mu_1, \mu_2) &\geq \min_{(\lambda,\mu_1,\mu_2)\in\Lambda} \max_{q=1,\ldots,N} \mathcal{L}(x^q, y^q, z^q, \lambda, \mu_1, \mu_2) \\ &= \gamma^N\end{aligned}$$

which proves the lemma. □

Lemma 4.3. *Let $N \geq 1$. Then β^N is an upper bound on the optimal solution value of the Lagrangian relaxation (7).*

Proof. As for all $(x, y, z) \in X$, $\mathcal{L}(x, y, z, \cdot, \cdot, \cdot)$ is linear and for all $(\lambda, \mu_1, \mu_2) \in \Lambda$ $\mathcal{L}(\cdot, \cdot, \cdot, \lambda, \mu_1, \mu_2)$ is linear and X as well as Λ are compact convex sets, it follows by von Neumanns's minimax theorem ([26],[31]) that

$$\min_{(\lambda,\mu_1,\mu_2)\in\Lambda} \max_{(x,y,z)\in X} \mathcal{L}((x, y, z, \lambda, \mu_1, \mu_2) = \max_{(x,y,z)\in X} \min_{(\lambda,\mu_1,\mu_2)\in\Lambda} \mathcal{L}(x, y, z, \lambda, \mu_1, \mu_2)$$

As

$$\begin{aligned}\beta^N &= \max_{(x,y,z)\in X} \min_{q=0,\ldots,N-1} \mathcal{L}(x, y, z, \lambda^q, \mu_1^q, \mu_2^q) \\ &\geq \max_{(x,y,z)\in X} \min_{(\lambda,\mu_1,\mu_2)\in\Lambda} \mathcal{L}(x, y, z, \lambda, \mu_1, \mu_2)\end{aligned}$$

the lemma is proved. □

As a direct consequence of proposition 4.1 and lemma 4.3 we get that the optimal solution value of problem LGNs(N) not only provides an upper bound on the optimal solution value of the Lagrangian relaxation (7) but also on the optimal solution value of GLCP (6) and thus of the initial SLBP (1) in case of discretely distributed random variables:

Corollary 4.4. *Let $N \geq 1$. Then β^N is an upper bound on the optimal solution value of GLCP (6).*

4.2 Stopping criteria

We define an absolute and a relative stopping criterion. Given $\delta > 0$ and $\epsilon > 0$, the iterative scheme stops if the absolute error is less than δ, i.e.

$$\beta^N - \gamma^N < \delta \text{ or } \beta^N - \gamma^{N-1} < \delta$$

or if the relative error is less than ϵ, i.e.

$$\frac{(\beta^N - \gamma^N)}{\beta^N} < \epsilon \text{ or } \frac{(\beta^N - \gamma^{N-1})}{\beta^N} < \epsilon$$

But does this case automatically arise when one of the problems have found an optimal solution? I.e., can we immediately detect that an optimal solution has been found? The answer is yes:

Lemma 4.5. *Let $N \geq 0$ and suppose that γ^N is the optimal solution value of the Lagrangian relaxation (7). Then β^{N+1} is also optimal with corresponding solution vector $(x^{N+1}, y^{N+1}, z^{N+1}, \lambda^N, \mu_1^N, \mu_2^N)$.*

Proof. If γ^N is optimal for problem (7), we have

$$(\lambda^N, \mu_1^N, \mu_2^N) = \arg \min_{(\lambda,\mu_1,\mu_2)\in\Lambda} \left(\max_{(x,y,z)\in X} \mathcal{L}(x, y, z, \lambda, \mu_1, \mu_2) \right)$$

and

$$\gamma^N = \max_{(x,y,z)\in X} \mathcal{L}(x, y, z, \lambda^N, \mu_1^N, \mu_2^N)$$

As $\beta^{N+1} \leq \mathcal{L}(x^{N+1}, y^{N+1}, z^{N+1}, \lambda^q, \mu_1^q, \mu_2^q)$ for all $q \leq N$ by constraints (8a), we get

$$\beta^{N+1} \leq \max_{(x,y,z)\in X} \mathcal{L}(x, y, z, \lambda^N, \mu_1^N, \mu_2^N) = \gamma^N$$

It has been shown in lemmata 4.2 and 4.3 that γ^N is a LB and β^{N+1} an UB and on the Lagrangian relaxation (7). It follows $\beta^{N+1} = \gamma^N$ which terminates the proof. □

Lemma 4.6. *Let $N \geq 0$ and suppose that β^N is the optimal solution value of the Lagrangian relaxation (7). Then γ^N is also optimal with corresponding solution vector $(x^N, y^N, z^N, \lambda^N, \mu_1^N, \mu_2^N)$.*

Proof. The proof is similar to the previous one. □

Therefore, we know directly if the current solution β^N or γ^N is optimal.

4.3 Convergence of the algorithm

In this subsection we will study the convergence of the minmax scheme. We will therefore suppose that GLCP (6) has a finite optimal solution value.
As in every iteration we add a new constraint to both problems, the following two lemmata follow immediately:

Lemma 4.7. *Let $N \geq 0$. Then $\beta^{N+1} \leq \beta^N$.*

□

Lemma 4.8. *Let $N \geq 1$. Then $\gamma^{N+1} \geq \gamma^N$.*

□

Lemmata 4.9 and 4.10 show that the minmax scheme is acyclic, i.e. at every iteration we either find an optimal solution of the Lagrangian relaxation (7) or generate two new feasible triples (x, y, z) and (λ, μ_1, μ_2):

Lemma 4.9. *Let $N \geq 2$. Then either $(\lambda^N, \mu_1^N, \mu_2^N) \neq (\lambda^h, \mu_1^h, \mu_2^h)$ for all $1 \leq h < N$ or γ^N is the optimal solution value of the Lagrangian relaxation (7).*

Proof. Let us assume that $(\lambda^N, \mu_1^N, \mu_2^N) = (\lambda^h, \mu_1^h, \mu_2^h)$ for a $h < N$. Then

$$\begin{aligned}
\gamma^N &= \max_{q \in \{1,\ldots,N\}} \mathcal{L}(x^q, y^q, z^q, \lambda^N, \mu_1^N, \mu_2^N) \\
&\geq \mathcal{L}(x^{h+1}, y^{h+1}, z^{h+1}, \lambda^N, \mu_1^N, \mu_2^N) \\
&= \mathcal{L}(x^{h+1}, y^{h+1}, z^{h+1}, \lambda^h, \mu_1^h, \mu_2^h) \\
&\geq \min_{q \in \{0,\ldots,h\}} \mathcal{L}(x^{h+1}, y^{h+1}, z^{h+1}, \lambda^q, \mu_1^q, \mu_2^q) = \beta^{h+1}
\end{aligned}$$

As γ^N is a lower bound on the Lagrangian relaxation (7) and β^{h+1} is an upper bound, it follows that $\gamma^N = \beta^{h+1}$ and γ^N (resp. β^{h+1}) is thus the optimal solution value of the Lagrangian relaxation. □

Lemma 4.10. *Let $N \geq 1$. Then either $(x^N, y^N, z^N) \neq (x^h, y^h, z^h)$ for all $0 \leq h < N$ or β^N is the optimal solution value of the Lagrangian relaxation (7).*

Proof. The proof is similar to the previous one. □

5 Modified iterative minmax scheme with convergence guarantee

As the Lagrangian relaxation (7) is continuous, it might theoretically be possible that we get stuck on a non-optimal solution value for infinitely many iterations, i.e. that there exists an $N^0 \geq 2$ such that either

$\mathcal{L}(x^{N^0}, y^{N^0}, z^{N^0}, \lambda^{N^0-1}, \mu_1^{N^0-1}, \mu_2^{N^0-1})$ is non-optimal, and for all $N \geq N^0$,

$\mathcal{L}(x^N, y^N, z^N, \lambda^{N-1}, \mu_1^{N-1}, \mu_2^{N-1}) = \mathcal{L}(x^{N^0}, y^{N^0}, z^{N^0}, \lambda^{N^0-1}, \mu_1^{N^0-1}, \mu_2^{N^0-1})$

or

$\mathcal{L}(x^{N^0}, y^{N^0}, z^{N^0}, \lambda^{N^0}, \mu_1^{N^0}, \mu_2^{N^0})$ is non-optimal and for all $N \geq N^0$,

$\mathcal{L}(x^N, y^N, z^N, \lambda^N, \mu_1^N, \mu_2^N) = \mathcal{L}(x^{N^0}, y^{N^0}, z^{N^0}, \lambda^{N^0}, \mu_1^{N^0}, \mu_2^{N^0})$.

However, by lemmata 4.5 and 4.6 we can immediately detect such a case as whenever we have $\beta^N = \beta^{N+1}$ ($\gamma^N = \gamma^{N+1}$) but $\gamma^N \neq \beta^{N+1}$ we can conclude that β^N and β^{N+1} (γ^N and γ^{N+1}) are non-optimal. In order to handle such cases, we propose the following modified minmax scheme: Whenever the case $\beta^N = \beta^{N+1}$ ($\gamma^N = \gamma^{N+1}$) arises we assure that the next upper (lower) bound produced is better than the previous one by at least δ by adding the following constraint:

$$\beta \leq \beta^N - \delta \quad \text{(respectively} \quad \gamma \geq \gamma^N + \delta \text{)} \tag{12}$$

Here δ is the absolute error that defines our stopping criterion, i.e. our algorithm stops when $\beta^N - \gamma^N < \delta$. So whenever adding a constraint of type (12) leads to an infeasible problem, we know that the last upper (lower) bound found has at most a difference of δ to the optimal solution value and we can immediately stop the iterations.

Let $LGNs(\mathbf{x}, \mathbf{y}, \mathbf{z}, \lambda, \mu_1, \mu_2, UB)$ be the function that solves subproblem (8). If the corresponding problem has a feasible solution, the function returns the optimal solution value. Otherwise, it returns a predefined value NF. Here (λ, μ_1, μ_2) are parameters that add a new constraint of the form (8a). This constraint is kept in the following calls of LGNs.

The function will store the solution vectors in the variables (x, y, z). If the problem has no feasible solution, (x, y, z) keep their input values. The parameter UB defines an additional constraint of type (12). This constraint is only kept for this single iteration. If $UB = \infty$ no constraint of type (12) is added.

In the same manner (just changing the roles of parameters and variables and the sign of the added constraint) we define $LGNd(x, y, z, \lambda, \boldsymbol{\mu_1}, \boldsymbol{\mu_2}, LB)$ with $LB = -\infty$ meaning that no additional constraint is added.

The proposed modified minmax scheme is stated in Algorithm 5.1. The variables UB^* and LB^* store the best upper and lower bound found so far while UB and LB are the bounds which we make the next calls of our problem solving functions with. tmp_{UB} and tmp_{LB} serve to store the return of these calls. The stopping criteria defined in section 4.2 are also used for this modified scheme.

6 Numerical experiments

We have performed computational experiments to test the convergence of the scheme proposed in section 5. To do so, we implemented the scheme presented in this paper and applied it on data randomly generated to match the general models at hand. In the results that follow, we have chosen not to present bound values, because the optimal values to compare them with are unknown. As a matter of fact, there is no out-of-the-box available solver to find the optimal solution of a given bilevel instance. The purpose of these experiments is to assess the convergence of the scheme we propose.

6.1 Data Generation

We randomly generate instances following the LBP model. Note that this is tantamount to generating data for the initial model SLBP since both problems are equivalent under the assumption that Ω is finite.

Although there exist a method to generate linear bilevel instances for which the optimal solution is known (see [25]), this method cannot produce instances of a specific type, e.g. fitting the LBP model, due to the solution construction mechanism.

The parameters of the data generation are the number of variables (resp. constraints) for the upper-level problem n_x (resp. m_1), the number of variables (resp. constraints) for the lower-level problem n_y (resp. m_2), the number of scenarii K and the risk α.

In order to generate a bounded polyhedron, we use the method described in [18] to set the components of A, B and b: for every row except the last one, A and B elements are uniformly picked in $[-1, 1]$. Components of the last row are uniformly chosen in $[0, 1]$. b is computed in the following manner:

$$b_i = \sum_{j=1}^{n} A_{ij} + B_{ij} + \rho,\ i = 1, \ldots, m$$

where ρ is uniformly chosen in $[0, 2]$. This ensures that the polyhedron defined by both set of constraints (4a) and (4h) is non-empty and bounded.

Constraints (4b) are generated as follows: for every scenario k, components of w_k are uniformly chosen in $[0, 1]$. Let $W_k = w_k^t \mathbb{1}_{n_x}$. We uniformly generate s_k in $[\frac{1}{2}W_k, W_k]$. The upper bound W_k is chosen so that scenario k is not necessarily respected. The lower bound $\frac{1}{2}W_k$ is arbitrarily chosen so that scenario k is not too restrictive. We set $M_k = W_k - s_k$ so that $z_k = 1$ disables the constraint.

The vector p of constraint (4c) respects:

$$\begin{cases} 0 \leq p_k \leq 1,\ k = 1, \ldots, K \\ p^t \mathbb{1}_K = 1 \end{cases}$$

where p_k is the probability of scenario k. The coefficients of the objective functions, i.e. components of c and d, are uniformly generated in $[0, 10]$. This generation procedure ensures that the feasible region of the LBP instance generated is bounded, but it does not guarantee non-emptiness.

Parameters			Time(s)			#LP		
K	n_x	n_y	Min	Avg	Max	Min	Avg	Max
20	50	50	0	0	1	17	29	57
		100	0	0	1	18	32	44
		500	11	13	17	17	24	32
		1000	113	137	172	16	20	27
	100	50	0	0	1	18	31	48
		100	0	0	2	23	49	84
		500	18	21	25	23	34	52
		1000	152	180	213	20	28	37
	500	50	5	7	10	13	26	43
		100	10	13	17	22	36	51
		500	95	123	179	46	81	177
		1000	431	549	795	53	82	116
	1000	50	46	61	76	13	25	37
		100	73	89	108	21	31	43
		500	293	383	458	34	57	89
		1000	854	1211	1549	32	75	129
50	50	50	0	0	1	15	28	56
		100	0	0	1	17	31	49
		500	13	15	18	18	25	31
		1000	115	134	166	15	20	26
	100	50	0	0	1	23	32	53
		100	0	0	2	27	44	73
		500	19	23	29	26	36	56
		1000	149	179	212	20	28	41
	500	50	6	8	12	13	28	47
		100	11	14	19	24	36	58
		500	89	127	199	39	80	213
		1000	446	535	652	47	80	151
	1000	50	50	64	78	10	26	41
		100	78	93	114	23	32	43
		500	277	389	629	25	56	89
		1000	638	1259	2235	10	77	113
100	50	50	0	0	1	19	28	45
		100	0	0	1	18	30	44
		500	13	15	19	16	23	32
		1000	115	140	166	16	20	26
	100	50	0	0	1	17	32	49
		100	1	1	2	29	42	79
		500	19	23	29	23	33	46
		1000	156	180	213	21	27	37
	500	50	7	9	14	15	27	45
		100	11	16	22	19	36	61
		500	102	130	169	50	77	151
		1000	485	548	720	53	79	163
	1000	50	50	67	92	17	26	40
		100	79	97	126	19	33	47
		500	303	405	760	37	56	77
		1000	867	1220	1528	33	77	122

Table 1: Convergence results of the iterative scheme.

6.2 Numerical Results

The iterative minmax scheme has been implemented in C++. Linear programs are solved with Cplex 11[1]. The absolute (resp. relative) tolerance is $\delta = 10^{-8}$ (resp. $\epsilon = 10^{-5}$). The maximum relative error ϵ has obviously a significant influence on the number of iterations needed to stop the scheme (see Figure 6.2). We have tested the scheme on instances generated using the procedure

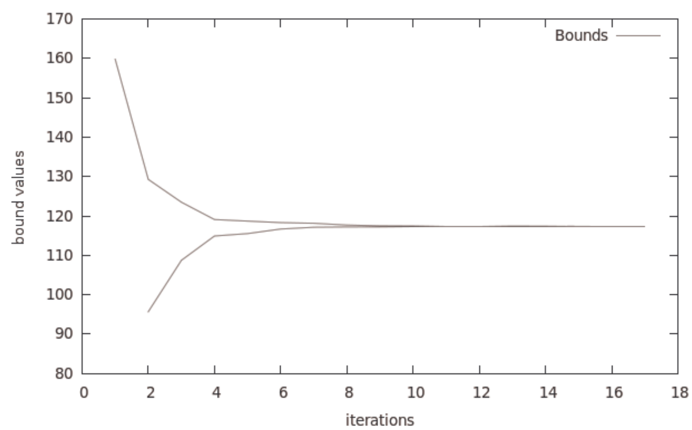

Figure 1: Plot of the bound values iteratively produced by the minmax scheme on one of the test instances randomly generated, with $n_x = n_y = K = 100$. The scheme stops after solving 33 linear programs, i.e. at the 17^{th} iteration. After 10 iterations, the relative error amounts to 0.14%. Upon reaching the stopping criteria, the relative error is 0.01%. This convergence behavior is recurrent.

described in 6.1. The scheme is initialized with a feasible solution of LGNd. Table 1 summarizes the results of the experiments. Results are only presented for $m_1 = n_x$ and $m_2 = n_y$. For each (K, n_x, n_y) triple, the scheme has been tested on 50 instances. For each triple, the minimum, average, maximum time (in seconds), number of Linear Programs solved (LGNs and LGNd added) are provided. The numerical tests confirm the theoretical results obtained in the previous sections, i.e. the iterative scheme reaches a near-optimal solution of LGN after finitely many steps. Not surprisingly, the time needed to reach the stopping criteria increases both with the size of the upper-level and lower-level problems, n_x and n_y. However, the computing time as well as the number of LP solved does not vary with the number of scenarii K. Also, a very interesting phenomenon arises once n_y becomes larger than n_x: the average number of LP iterations decreases as n_y increases (see for example the first four valued rows of column *Avg* #LP). The explanation is intuitive: the bigger the lower-level problem, the larger the Lagrangian $\mathcal{L}(x, y, z, \lambda, \mu_1, \mu_2)$, and therefore the deeper

[1]Computation times have been measured on *NEC Express5800 120Rh (4*2GHz,3GB)* servers, with two single-processed schemes simultaneously running.

each cut is. Deeper cuts decrease the number of iterations needed to reach the stopping criteria.
The parameter modeling the risk, α, is set to 0.1 in the test instances presented here. The scheme also converges for other α values, with little to no variations in the computing time and the number of LP solved. These numerical results confirm the convergence properties presented in section 4.

7 Conclusion

We study a novel stochastic bilevel problem with probabilistic knapsack constraints, which can be used to jointly optimize network resources and service pricing. We provide a rich application to network optimization. The initial problem is transformed into an equivalent quadratic problem, which, in turn, is relaxed into a linear minmax problem. An iterative approach with convergence guarantee applied to the latter allows us to find upper bounds of the original stochastic bilevel problem. Numerical experiments confirm that the method we propose converges. Further reasearch includes using the scheme to provide bounds for a Branch and Bound algorithm.

References

[1] F.A. Al-Khayyal, *Jointly constrained bilinear programs and related problems: An overview*, Computers & Mathematics with Applications **19** (1990) pp. 53–62.

[2] F. A. Al-Khayyal and J. E. Falk, *Jointly constrained biconvex programming*, Mathematics of Operations Research **8** (1983) pp. 273–286.

[3] J. A. Audestad, A. A. Gaivoronski and A. Werner, *Extending the stochastic programming framework for the modeling of several decision makers: Pricing and competition in the telecommunication sector*, Annals of Operations Research, **142** (2006) pp. 19–39.

[4] C. Audet, P. Hansen, B. Jaumard and Gilles Savard, *Links Between Linear Bilevel and Mixed 0-1 Programming Problems*, Journal of optimization theory and applications **2** (1997) pp. 273–300.

[5] M. Bouhtou, G. Erbs and M. Minoux, *Joint optimization of pricing and resource allocation in competitive telecommunications networks*, Networks **1** (2007) pp. 37–49.

[6] M. Bouhtou, S. P. M. van Hoesel, A. F. van der Kraaij and J.-L. Lutton, *Tariff optimization in networks*, Journal of Computing **19** (2007) pp. 458–469.

[7] L. Brotcorne, P. Marcotte and G. Savard, *Bilevel Programming: The Montreal School*, INFOR: Information Systems and Operational Research **46** (2008) pp. 219–312.

[8] S. Christiansen, M. Patriksson and L. Wynter, *Stochastic bilevel programming in structural optimization* Structural and Multidisciplinary Optimization **21** (2001) pp. 361–371.

[9] B. Colson, P. Marcotte and G. Savard, *An overview of bilevel optimization*, Annals of Operations Research **153** (2007) pp. 235–256.

[10] J.-P. Côté, P. Marcotte and G. Savard, *A bilevel modeling approach to pricing and fare optimization in the airline industry*, Journal of Revenue and Pricing Management **2** (2003) pp 23–36.

[11] S. Dempe and K. Richter, *Bilevel Programming With Knapsack Constraints*, Central European Journal of Operations Research **8(2)** (2000) pp. 93–107.

[12] S. Dempe, *Foundations of Bilevel Programming*, Kluwer Academic Publishers, Dordrecht, 2002.

[13] J.E. Falk, *A linear max-min problem*, Mathematical Programming **5** (1973) pp. 169–188.

[14] C.A. Floudas and V. Visweswaran, *Quadratic Optimization*, Handbook of Global Optimization (R.Horst and P.M. Pardalos, Editors) (1994) Kluwer Academic Publishers, pp. 217–270.

[15] J. Fortuny-Amat and B. McCarl, *A Representation and Economic Interpretation of a Two-Level Programming Problem* Journal of the Operational Research Society **32** (1981) pp. 783–792.

[16] A. A. Gaivoronski, *Stochastic optimization in telecommunications* Handbook of Optimization in Telecommunications (M.G. C. Resende and P. M. Pardalos, Editors) (2006), Springer-Verlag, chapter 27, pp. 761–799.

[17] G. Heiporn, M. Labbe, P. Marcotte and G. Savard, *A polyhedral study of the network pricing problem with connected toll arcs*, Networks **55** (2009) pp. 234–246

[18] R. Horst and N. V. Thoai, *Modification, implementation and comparison of three algorithms for globally solving linearly constrained concave minimization problems*, Computing **42** (1989) pp. 271–289.

[19] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd ed. Berlin: Springer-Verlag, 1996.

[20] H. Konno, *A cutting plane algorithm for solving bilinear programs*, Mathematical Programming **11** (1976) pp. 14–27.

[21] S. Kosuch and A. Lisser, *Upper bounds for the 0-1 stochastic knapsack problem and a B&B algorithm*, Annals of Operations Research **176** (2010) pp. 77–93.

[22] M. Labb, P. Marcotte and G. Savard *A bilevel model of taxation and its applications to optimal highway pricing* Management Science **44** (1998) pp. 1595–1607.

[23] P. B. Luh, T. S. Chang and T. Ning, *Pricing problems with a continuum of customers as stochastic Stackelberg games*, Journal of Optimization Theory and Applications **55** (1987) pp. 119–131.

[24] D. Mitra, K.G. Ramakrishnan and Q. Wang, *Combined economic modeling and traffic engineering: Joint optimization of pricing and routing in multi-service networks*, Proceedings of 17th International Teletraffic Congress, 2001. Elsevier.

[25] K. Moshirvaziri, M. A. Amouzegar and S. E. Jacobsen, *Test problem construction for linear bilevel programming problems*, Journal of Global Optimization **8** (1996) pp.235–243.

[26] J. von Neumann, *Zur Theorie der Gesellschaftsspiele*, Math. Annalen **100** (1928) pp. 295–320.

[27] O. Y. Özaltin, O. A. Prokopyev and A. J. Schaefer, *The Bilevel Knapsack Problem with Stochastic Right-Hand Sides*, Operations Research Letters **38(4)** (2010) pp. 328–333.

[28] V. M. Panin, *Finite algorithm to find the saddle point of a quadratic function subject to linear constraints*, Cybernetics and Systems Analysis **1** (1993) pp. 47–61.

[29] M. Patriksson and L. Wynter *Stochastic mathematical programs with equilibrium constraints* Operations Research Letters **25** (1999) pp. 159–167

[30] H.D. Sherali and C. M. Shetty, *A finitely convergent algorithm for bilinear programming problems using polar cuts and disjunctive face cuts*, Mathematical Programming **19** (1980) pp. 14–31.

[31] M. Sion, *On general minimax theorems* , Pacific Journal of Mathematics **81** (1958) pp. 171–176.

[32] L. Vicente, Gilles Savard and J. Judice, *Discrete Linear Bilevel Programming Problem*, Journal of optimization theory and applications **3** (1996) pp. 597–614.

[33] A. S. Werner, *Bilevel stochastic programming problems: analysis and application to telecommunications*, PhD thesis, Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology.

Iterative minmax Algorithm

```
Require: ∃(x, y, z, λ, μ1, μ2) feasible for the Lagrangian relaxation (7)
Require: δ > 0, ϵ > 0
  UB ← ∞, UB* ← ∞
  LB ← −∞, LB* ← −∞
  (λ, μ1, μ2) feasible for the Lagrangian relaxation (7)
  loop
    tmp_UB ← LGNs(x, y, z, λ, μ1, μ2, UB)
    if tmp_UB = NF then
      LB* = UB
      BREAK LOOP
    else
      if tmp_UB = UB* then
        UB ← tmp_UB − δ
      else
        UB* ← tmp_UB
        UB ← ∞
        if UB* − LB* < δ or (UB* − LB*)/UB* < ϵ then
          BREAK LOOP
        end if
      end if
    end if

    tmp_LB ← LGNd(x, y, z, λ, μ1, μ2, LB)
    if tmp_LB = NF then
      UB* = LB
      BREAK LOOP
    else
      if tmp_LB = LB* then
        LB ← tmp_LB + δ
      else
        LB* ← tmp_LB
        LB ← −∞
        if UB* − LB* < δ or (UB* − LB*)/UB* < ϵ then
          BREAK LOOP
        end if
      end if
    end if
  end loop
  return LB* and/or UB*
```

Algorithm 5.1: The iterative minmax scheme with convergence guarantee.