

On a class of stochastic programs with exponentially many scenarios

Gustavo Angulo
Pontificia Universidad Católica de Chile
gangulo@ing.puc.cl

April 1, 2020

Abstract

We consider a class of stochastic programs whose uncertain data has an exponential number of possible outcomes, where scenarios are affinely parametrized by the vertices of a tractable binary polytope. Under these conditions, we propose a novel formulation that introduces a modest number of additional variables and a class of inequalities that can be efficiently separated. Moreover, when the underlying polytope is the unit hypercube, we present an extended formulation of polynomial size that can be solved directly with off-the-shelf optimization software. We assess the advantages and limitations of our formulation through a computational study.

1 Introduction

In stochastic programming, it is usually assumed that the random data of the problem follows a known distribution \mathbb{P} . When \mathbb{P} is either continuous or finite with a large number of atoms, sampling methods can be used to approximate the true problem by a model involving a reasonable number of scenarios. But what happens when \mathbb{P} is “easy” to describe and still involves an enormous number of possible outcomes? A natural question to ask is whether we can solve the true problem without relying on sampling methods. One such an example are models with exponentially many scenarios. For instance, scenarios might be given by all subsets of fixed cardinality drawn from a finite ground set. Work along this line is, however, rather limited. In [6], robust combinatorial problems of this type are studied for which approximation algorithms are developed. Further results are obtained in [8]. Below we present a model that illustrates a potential application of implicit representation of scenarios in the context of chance-constrained problems (CCP) and discuss some relevant features and questions.

Consider an optimization problem where the data defining the constraints is uncertain. Suppose that the objective of the decision-maker is to find a solution that is feasible with high probability over the possible realization of the data, while incurring the least possible cost. This problem can be formulated as a CCP of the form

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & \mathbb{P}(A(\tilde{\omega})x \geq b(\tilde{\omega})) \geq 1 - \epsilon \\ & x \in X, \end{aligned}$$

where $X \subseteq \mathbb{R}^n$ imposes deterministic constraints on the decision vector x , $c \in \mathbb{R}^n$ defines a linear objective function, $(A(\tilde{\omega}), b(\tilde{\omega}))$ denotes the uncertain data, and $\epsilon > 0$ is the failure tolerance.

CCPs with independent constraints were introduced in [5], while jointly-constrained problems were considered in [13]. Although CCPs appear in a number of applications, unfortunately they are hard to solve in general due to the nonconvexity of the feasible set. Sampling methods that provide safe approximations and statistical guarantees include [3, 4, 15, 16, 11].

We restrict our attention to the case where the linear system has a single constraint having a finite set of possible realizations parametrized by a left-hand-side vector $a \in \mathcal{A} \subseteq \mathbb{R}^n$. We are asked for a solution vector x that satisfies all the constraints given by \mathcal{A} with the exception of at most $k - 1$ constraints. That is, we want to solve

$$\begin{aligned} (\text{P}_{k-1}) \quad & \min \quad c^\top x \\ & \text{s.t.} \quad a^\top x \geq b \quad \text{for all but at most } k - 1 \text{ vectors } a \in \mathcal{A} \\ & \quad \quad x \in X. \end{aligned}$$

This particular model is known as $(k - 1)$ -violation linear programming (LP) problem and was introduced in [18]. Note that (P_{k-1}) can be seen as a CCP with \mathbb{P} being the uniform distribution over \mathcal{A} and $k - 1 = \lfloor \epsilon |\mathcal{A}| \rfloor$. A standard integer programming (IP) formulation for this problem includes binary variables z_a to decide whether a particular constraint is satisfied or not. It has the form

$$\begin{aligned} (\text{IP}_{\text{std}}) \quad & \min \quad c^\top x \\ & \text{s.t.} \quad a^\top x + M_a z_a \geq b \quad \forall a \in \mathcal{A} \\ & \quad \quad \sum_{a \in \mathcal{A}} z_a \leq k - 1 \\ & \quad \quad x \in X \\ & \quad \quad z_a \in \{0, 1\} \quad \forall a \in \mathcal{A}, \end{aligned}$$

where $M_a > 0$ is a sufficiently large constant such that the constraint $a^\top x + M_a z_a \geq b$ is satisfied for any $x \in X$. In [17] several enhancements over a commercial solver are presented for covering problems of this type.

Now, suppose that \mathcal{A} is extremely large, say exponential in n , and thus listing all the variables and constraints in (IP_{std}) is impractical.

Example 1. Let $a \in \{0, 1\}^n$ represent the set of resources that are available in a given scenario. If at least p of them are available in any scenario, then $\mathcal{A} = \{a \in \{0, 1\}^n : \sum_{i=1}^n a_i \geq p\}$. Thus $|\mathcal{A}| = \sum_{j=p}^n \binom{n}{j}$ and in particular $|\mathcal{A}| = 2^{n-1}$ for $p = n/2$.

In general, the following issues arise:

- The number of variables and constraints in (IP_{std}) is extremely large. Therefore, the natural IP formulation might turn unmanageable due to its size.
- The IP formulation makes use of “big- M ” constraints, which are known to lead to weak LP relaxations. Therefore, solving (IP_{std}) might take a considerable amount of time.

- Although decomposition methods such as column–and–row generation techniques [14, 19] could be used to incrementally add variables and constraints, these methods are incompatible with the enhancements in [17] since in the latter the complete formulation must be available a priori. Therefore, decomposition methods of this type are likely to suffer from weak LP relaxations due to “big- M ” constraints.

Hence, it is desirable to have at hand alternative formulations that allow for efficient solutions methods. In this work, we make the following contributions towards this end:

- We introduce a novel reformulation for (P_{k-1}) which does not require binary variables for all scenarios. Under a parametric condition on \mathcal{A} , this formulation introduces a modest number of additional variables. Unlike [6] and [8], our setting is rather general and not tailored to specific classes of combinatorial problems.
- Under a tractability condition, we show that the constraints in our formulation are easy to separate. Moreover, when the parametrization is given by the unit hypercube, we show how to construct a compact extended IP formulation.
- We assess the advantages and limitations of our formulation through a series of computational experiments.

The remainder of the paper is organized as follows. In Section 2, we present our assumptions and the general approach. In Section 3, we derive new formulations for (P_{k-1}) , and some extensions are discussed in Section 4. Our computational experiments are reported in Section 5 and concluding remarks in Section 6.

2 The approach

In many instances of (P_{k-1}) , the description of the problem, in particular that of \mathcal{A} , may be fairly simple and yet lead to a complicated IP formulation. Continuing with Example 1, note that the encoding of \mathcal{A} is rather straightforward, we just need n and p , but the IP formulation is very large. Given these observations, we propose to use an implicit formulation to solve problems of this type. To that end, let \mathcal{S} be the family of $(k-1)$ -sets of elements in \mathcal{A} . We are interested in solving

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & a^\top x \geq b \quad \forall a \in \mathcal{A} \setminus S \\ & x \in X \\ & S \in \mathcal{S}, \end{aligned}$$

that is, we have to decide on the vector x and the discarded set S at the same time without enumerating all the variables and constraints implied by the set of scenarios \mathcal{A} .

Let $\phi(x, S) := \min\{x^\top a : a \in \mathcal{A} \setminus S\}$. Also, for fixed x , let $S(x) \in \mathcal{S}$ denote the set of realizations of a that yield the $k-1$ smallest values of $x^\top a$. With $v_k(x) := \phi(x, S(x))$ being the k -th smallest value of $x^\top a$ over \mathcal{A} , the problem is equivalent to

$$\begin{aligned} (P_v) \quad \min \quad & c^\top x \\ \text{s.t.} \quad & v_k(x) \geq b \\ & x \in X. \end{aligned}$$

With this formulation, the question becomes whether $v_k(x)$ can be efficiently computed and/or represented or not. Below we make two assumptions on this regard.

Assumption 1. \mathcal{A} is parametrized by an affine function over the vertices of a polytope. More precisely, there exist a polytope $Q \subseteq \mathbb{R}^m$, a matrix $A \in \mathbb{R}^{n \times m}$ and a vector $\bar{a} \in \mathbb{R}^n$ such that, for any $a \in \mathcal{A}$, there is a unique $z \in \text{vert}(Q)$ satisfying $a = \bar{a} + Az$, where $\text{vert}(Q) \subseteq Q$ is the set of vertices of Q .

Assumption 1 let us encode a large number of scenarios implicitly via an affine transformation over a polytope.

Example 2. Suppose we have n independent assets which can yield gains ($g_j > 0$) or losses ($l_j < 0$). If \mathcal{A} is the set of uncertain return vectors $a \in \mathbb{R}^n$, we have

$$\begin{aligned} \mathcal{A} &= \{a \in \mathbb{R}^n \mid a_j \in \{l_j, g_j\} \forall i = 1, \dots, n\} \\ &= \{a \in \mathbb{R}^n \mid \exists z \in \{0, 1\}^n : a_j = l_j + (g_j - l_j)z_j \forall i = 1, \dots, n\} \\ &= \{a \in \mathbb{R}^n \mid \exists z \in \text{vert}([0, 1]^n) : a = l + \text{diag}(g - l)z\}. \end{aligned}$$

Taking $m = n$, $Q = [0, 1]^m$, $\bar{a} = l$ and $A = \text{diag}(g - l)$, we obtain an affine parametrization of \mathcal{A} as in Assumption 1.

Given $x \in X$, in order to check if $v_k(x) \geq b$ holds, we have to compute the k -th smallest value of $x^\top a$ over $a \in \mathcal{A}$, or equivalently, the k -th smallest value of $x^\top (\bar{a} + Az)$ over $z \in \text{vert}(Q)$. This problem reduces to computing the k best basic solutions to an LP, which is NP-hard in general [2]. Therefore, for tractability, we restrict our attention to polytopes for which computing the k best basic solutions with respect to a given linear objective function can be done efficiently. This is the case, for instance, of 0/1 polytopes associated to polynomially-solvable combinatorial problems [9, 2].

Assumption 2. Q is a tractable binary polytope. More precisely, $\text{vert}(Q) \subseteq \{0, 1\}^m$ and both separation and linear optimization over Q can be done in polynomial time with respect to m .

Under Assumption 2, $v_k(x)$ is easy to compute. Unfortunately, this function is nonconvex and non-concave in general.

Now, let $\sigma_l(x)$ denote the sum of the l smallest values of $x^\top a$ over $a \in \mathcal{A}$, which is a concave function of x . We have $v_k(x) = \sigma_k(x) - \sigma_{k-1}(x)$, obtaining

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & \sigma_k(x) - \sigma_{k-1}(x) \geq b \\ & x \in X. \end{aligned}$$

Let $\bar{\sigma}_k(x) := \frac{1}{k}\sigma_k(x)$ and note that $\bar{\sigma}_k(x) \leq v_k(x)$ holds. Therefore, $\bar{\sigma}_k(x) \geq b$ is a convex constraint and the more restrictive problem

$$\begin{aligned} (P_{\bar{\sigma}}) \quad \min \quad & c^\top x \\ \text{s.t.} \quad & \bar{\sigma}_k(x) \geq b \\ & x \in X \end{aligned}$$

provides an upper bound for (P_v) . Moreover, observe that having constraints on $v_k(x)$ and $\bar{\sigma}_k(x)$ is analogous to having constraints on the Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) of a random variable, respectively. In fact, the CVaR at level $\alpha \in (0, 1)$ is related to the average of a random variable over the α upper tail of its distribution, which is exactly the same idea behind $\bar{\sigma}_k(x)$ with k playing the role of α .

3 Polynomial and exponential formulations

In this section, we address the representation of $\sigma_k(x)$ when Q is a binary polytope, which we latter use to represent $v_k(x)$ and reformulate (P_{k-1}) .

3.1 Representing $\sigma_k(x)$ with the unit hypercube

Let us assume that $Q = [0, 1]^m$. In this case, computing $\bar{\sigma}_k(x)$ amounts to finding the average of the k best solutions to $\min\{x^\top(\bar{a} + Az) : z \in \{0, 1\}^m\}$, which can be formulated as

$$\begin{aligned} \bar{\sigma}_k(x) = \min & \quad \frac{1}{k} \sum_{i=1}^k x^\top A z_i + x^\top \bar{a} \\ \text{s.t.} & \quad z_i \neq z_j \quad i \neq j \\ & \quad z_i \in \{0, 1\}^m \quad 1 \leq i \leq k. \end{aligned}$$

Although linear optimization over binary all-different polytopes can be done in polynomial time [2], no compact linear formulation in k and m is known. On the other hand, given that the objective $x^\top A$ is common to the k vectors z_i , we also have

$$\begin{aligned} \bar{\sigma}_k(x) = \min & \quad \frac{1}{k} x^\top A y + x^\top \bar{a} \\ \text{s.t.} & \quad \sum_{i=1}^k z_i = y \\ & \quad z_i \prec z_j \quad i < j \\ & \quad z_i \in \{0, 1\}^m \quad 1 \leq i \leq k, \end{aligned}$$

where \prec denotes the usual lexicographical order. In this case, we can represent the constraint $\bar{\sigma}_k(x) \geq b$ with polynomially-many additional variables and constraints as follows.

Let $L_{k,m}$ be the set of $k \times m$ binary matrices $z = [z_1, \dots, z_k]^\top$ with rows $z_1^\top, \dots, z_k^\top \in \{0, 1\}^m$ in increasing lexicographical order, and let $Y_{k,m} := \{y \in \mathbb{Z}_+^m : y = \sum_{i=1}^k z_i \text{ for some } z \in L_{k,m}\}$. Then the constraint $\bar{\sigma}_k(x) \geq b$ becomes

$$x^\top A y + k x^\top \bar{a} \geq k b \quad \forall y \in Y_{k,m}. \quad (1)$$

Instead of relying on separation to enforce (1), we seek a description of $\text{conv}(Y_{k,m})$ of polynomial size with respect to k and m and then apply LP duality to represent (1).

In [7], an extended formulation for $\text{conv}(L_{k,m})$ with $\mathcal{O}(k^3 m)$ variables and constraints is derived using the framework of branched polyhedral systems. Another formulation of the same size can be directly derived from a dynamic program that solves linear optimization over $L_{k,m}$ [10]. We present a dynamic program with complexity $\mathcal{O}(k^2 m)$ to optimize a linear function over $Y_{k,m}$, from which we derive an extended formulation for $\text{conv}(Y_{k,m})$ with $\mathcal{O}(k^2 m)$ variables and constraints.

Theorem 1. *There exist matrices E and F , and a vector g such that the system $Ey + Fw = g$, $w \geq 0$ has $\mathcal{O}(k^2 m)$ variables and constraints, and the projection of its solution set onto the y -space is equal to $\text{conv}(Y_{k,m})$.*

Proof. Given $d \in \mathbb{R}^m$, let $\gamma^* := \min\{d^\top y : y \in Y_{k,m}\} = \min\{d^\top \sum_{i=1}^k z_i : [z_1, \dots, z_k]^\top \in L_{k,m}\}$. For $1 \leq t \leq m$, let $d_{[t]} := (d_t, \dots, d_m) \in \mathbb{R}^{m-t+1}$, and for $1 \leq i \leq k$, let $\gamma_i^t := \min\{d_{[t]}^\top y : y \in Y_{i,m-t+1}\}$. With these definitions, we present a dynamic program to compute $\gamma^* = \gamma_k^1$. In what follows, $\mathbf{0}$ and $\mathbf{1}$ represent vectors of 0's and 1's of appropriate size, respectively.

Let $y \in Y_{k,m}$ be an optimal solution defining γ_k^1 , and let $z \in L_{k,m}$ be such that $y = \sum_{i=1}^k z_i$. Observe that the first column of z is either equal to $\mathbf{0}$ or $\mathbf{1}$, or it splits in a sequence of 0's followed by a sequence of 1's. In the former case, $z = [\mathbf{0} \bar{z}]$ or $z = [\mathbf{1} \bar{z}]$ for some $\bar{z} \in L_{k,m-1}$, and thus $y = (0, \bar{y})$ or $y = (k, \bar{y})$ for some $\bar{y} \in Y_{k,m-1}$. Therefore, if y is of this form, it will have objective value equal to $0 + \gamma_k^2$ or $kd_1 + \gamma_k^2$, whichever is the least, with γ_k^2 given by an optimal choice of \bar{y} . In the latter case, if $k > 1$, the split ensures that the first group of rows $[\mathbf{0} z']$ are lexicographically smaller than the second group $[\mathbf{1} z'']$, where $z' \in L_{l,m-1}$ and $z'' \in L_{k-l,m-1}$ for some $1 \leq l < k$, and thus $y = (0, y') + (k-l, y'') = (k-l, y' + y'')$ for some $y' \in Y_{l,m-1}$ and $y'' \in Y_{k-l,m-1}$. Therefore, if y is of this form, it will have objective value equal to the least of $0 + \gamma_l^2 + (k-l)d_1 + \gamma_{k-l}^2$ over $1 \leq l < k$, with γ_l^2 and γ_{k-l}^2 given by optimal choices of y' and y'' , respectively. This idea applies recursively to γ_i^t as long as $1 \leq t < m$. For $t = m$, we have $L_{1,1} = \{[0], [1]\}$ and $Y_{1,1} = \{(0), (1)\}$, $L_{2,1} = \{[0 \ 1]^\top\}$ and $Y_{2,1} = \{(1)\}$, and $Y_{i,1} = \emptyset = L_{i,1}$ for $i > 2$, yielding $\gamma_1^m = \min\{0, d_m\}$, $\gamma_2^m = 0 + d_m$ and $\gamma_i^m = +\infty$ for $i > 2$, respectively.

The above construction leads to the dynamic programming formulation (2a)–(2e) below to compute $\gamma^* = \gamma_k^1$:

- For $1 \leq t < m$ and $1 < i \leq k$:

$$\gamma_i^t = \min \left\{ 0 + \gamma_i^{t+1}, id_t + \gamma_i^{t+1}, \min_{1 \leq l < i} \left\{ 0 + \gamma_l^{t+1} + (i-l)d_t + \gamma_{i-l}^{t+1} \right\} \right\} \quad (2a)$$

- For $1 \leq t < m$ and $i = 1$:

$$\gamma_1^t = \min \left\{ 0 + \gamma_1^{t+1}, d_t + \gamma_1^{t+1} \right\} \quad (2b)$$

- For $t = m$:

$$\gamma_1^m = \min\{0, d_m\} \quad (2c)$$

$$\gamma_2^m = 0 + d_m \quad (2d)$$

$$\gamma_i^m = +\infty \quad \text{for } i > 2. \quad (2e)$$

Following the approach of [12], in order to find an extended formulation for $\text{conv}(Y_{k,m})$, we define a hypergraph H representing the dynamic program (2a)–(2e). The state space N is identified with variables γ_i^t . In order to define the hyperarcs in H , we first cast (2a)–(2e) in LP form, yielding the system (3a)–(3d) below:

- For $1 \leq t < m$ and $1 < i \leq k$:

$$\gamma_i^t \leq \begin{cases} \gamma_i^{t+1} & (p_i^t) \\ id_t + \gamma_i^{t+1} & (q_i^t) \\ \gamma_l^{t+1} + (i-l)d_t + \gamma_{i-l}^{t+1} & \text{for } 1 \leq l < i \quad (r_{il}^t) \end{cases} \quad (3a)$$

- For $1 \leq t < m$ and $i = 1$:

$$\gamma_1^t \leq \begin{cases} \gamma_1^{t+1} & (p_1^t) \\ d_t + \gamma_1^{t+1} & (q_1^t) \end{cases} \quad (3b)$$

- For $t = m$:

$$\gamma_1^m \leq \begin{cases} 0 & (p_1^m) \\ d_m & (q_1^m) \end{cases} \quad (3c)$$

$$\gamma_2^m \leq d_m \quad (r_{21}^m), \quad (3d)$$

where the objective is to maximize γ_k^1 . For each linear constraint in (3a)–(3b), we include in H an hyperarc of the form (Γ, γ_i^t) , where $\Gamma \subseteq \{\gamma_1^{t+1}, \dots, \gamma_i^{t+1}\}$ is the set of variables in the right-hand side, along with hyperarcs (\emptyset, γ_1^m) and (\emptyset, γ_2^m) from the boundary constraints (3c)–(3d). In this manner, H is acyclic.

From (3a)–(3d), by LP duality, we obtain $p, q, r \geq 0$ such that

- For $1 < i \leq k$:

$$p_i^t - p_i^{t-1} + q_i^t - q_i^{t-1} + \sum_{1 \leq l < i} r_{il}^t - \sum_{i < j \leq k} r_{ji}^{t-1} - \sum_{i < j \leq k} r_{j(j-i)}^{t-1} = 0 \quad \text{for } 1 < t < m \quad (4a)$$

$$p_i^1 + q_i^1 + \sum_{1 \leq l < i} r_{il}^1 = \begin{cases} 0 & \text{if } 1 < i < k \\ 1 & \text{if } i = k \end{cases} \quad \text{for } t = 1 \quad (4b)$$

$$-p_i^{m-1} - q_i^{m-1} - \sum_{i < j \leq k} r_{ji}^{m-1} - \sum_{i < j \leq k} r_{j(j-i)}^{m-1} = \begin{cases} -r_{21}^m & \text{if } 2 = i \\ 0 & \text{if } 2 < i \leq k \end{cases} \quad \text{for } t = m \quad (4c)$$

- For $i = 1$:

$$p_1^t - p_1^{t-1} + q_1^t - q_1^{t-1} - \sum_{1 < j \leq k} r_{j1}^{t-1} - \sum_{1 < j \leq k} r_{j(j-1)}^{t-1} = 0 \quad \text{for } 1 < t \leq m \quad (4d)$$

$$p_1^1 + q_1^1 = 0 \quad \text{for } t = 1. \quad (4e)$$

To complete the approach in [12], we define $I = \{1, \dots, k\} \times \{1, \dots, m\}$ and tag each state $\gamma_i^t \in N$ with reference set $I[\gamma_i^t] = \{1, \dots, i\} \times \{t+1, \dots, m\} \subseteq I$. These sets fulfill the consistency and disjointness conditions of [12], namely, for all $(\Gamma, \gamma_i^t) \in H$, we have $I[\gamma_i^{t+1}] \subseteq I[\gamma_i^t]$ for all $\gamma_i^t \in \Gamma$ and $I[\gamma_i^{t+1}] \cap I[\gamma_{i'}^{t+1}] = \emptyset$ for all distinct $\gamma_i^{t+1}, \gamma_{i'}^{t+1} \in \Gamma$. These conditions imply that the polyhedron (4a)–(4e) is integral with binary vertices only.

The objective function of the dual problem is to minimize

$$\sum_{1 \leq t < m} \left(\sum_{1 \leq i \leq k} id_t q_i^t + \sum_{1 \leq l < i \leq k} (i-l)d_l r_{il}^t \right) + d_m q_1^m + d_m r_{21}^m.$$

Grouping terms with respect to each d_t , we find that y is given by

$$y_t = \begin{cases} \sum_{1 \leq i \leq k} i q_i^t + \sum_{1 \leq l < i \leq k} (i-l) r_{il}^t & \text{for } 1 \leq t < m \\ q_1^m + r_{21}^m & \text{for } t = m. \end{cases} \quad (4f)$$

Taking $w := (p, q, r)$ and writing (4a)–(4f) as $Ey + Fw = g$, $w \geq 0$, we obtain the result as (4f) preserves integrality. \square \square

From Theorem 1, we can assume that we have at hand a system of polynomial size such that

$$k\bar{\sigma}_k(x) = \min\{x^\top Ay : Ey + Fw = g, w \geq 0\} + kx^\top \bar{a}.$$

By LP duality, we have

$$k\bar{\sigma}_k(x) = \max\{g^\top \pi : E^\top \pi = A^\top x, F^\top \pi \leq 0\} + kx^\top \bar{a}.$$

Therefore, $\bar{\sigma}_k(x) \geq b$ can be compactly represented by the system

$$\begin{aligned} g^\top \pi + k\bar{a}^\top x &\geq kb \\ E^\top \pi &= A^\top x \\ F^\top \pi &\leq 0. \end{aligned}$$

3.2 Representing $\sigma_k(x)$ with binary polytopes

Given a binary polytope $Q \subseteq \mathbb{R}^m$, we have that $Q \cap \{0, 1\}^m$ is precisely the set of vertices of Q , and thus

$$\begin{aligned} \bar{\sigma}_k(x) = \min & \frac{1}{k} \sum_{i=1}^k x^\top Az_i + x^\top \bar{a} \\ \text{s.t.} & z_i \neq z_j \quad i \neq j \\ & z_i \in Q \cap \{0, 1\}^m \quad 1 \leq i \leq k. \end{aligned}$$

Let $L_{k,Q} := L_{k,m} \cap Q^k$ and $Y_{k,Q} := \{y \in \mathbb{Z}_+^m : y = \sum_{i=1}^k z_i \text{ for some } [z_1, \dots, z_k]^\top \in L_{k,Q}\}$. Then, along the same lines of the case of the unit hypercube, $(P_{\bar{\sigma}})$ can be cast as

$$\begin{aligned} \min & c^\top x \\ \text{s.t.} & x^\top Ay + k\bar{a}^\top x \geq kb \quad \forall y \in Y_{k,Q} \\ & x \in X. \end{aligned} \tag{5}$$

Although it might not be possible to represent (5) in a compact manner, these inequalities can be efficiently separated, provided that Q is tractable.

Proposition 2. *Under Assumption 2, inequalities (5) can be separated in polynomial time.*

Proof. To separate (5) given $x \in X$, we must solve $\min\{x^\top Ay : y \in Y_{k,Q}\}$, which is equivalent to $\min\{x^\top A \sum_{i=1}^k z_i : [z_1, \dots, z_k]^\top \in L_{k,Q}\}$. The latter reduces to computing the k best basic solutions to $\min\{x^\top Az : z \in Q\}$, which can be done in polynomial time [9, 2]. \square \square

3.3 Representing $v_k(x)$ and completing the formulation

In order to represent the constraint $v_k(x) \geq b$, we write it as $\sigma_k(x) - b \geq \sigma_{k-1}(x)$ and consider any feasible solution to the problem defining the right-hand side. Since $\sigma_{k-1}(x) = \min\{\sum_{a \in S} a^\top x : S \in$

\mathcal{S} }, we have $\sigma_{k-1}(x) \leq x^\top A \sum_{i=1}^{k-1} z_i + (k-1)\bar{a}^\top x$ for any $z \in L_{k-1,m}$, or equivalently, $\sigma_{k-1}(x) \leq x^\top Ay + (k-1)\bar{a}^\top x$ for any $y \in Y_{k-1,m}$. We thus arrive at the system

$$\begin{aligned} \sigma_k(x) - b &\geq x^\top Ay + (k-1)\bar{a}^\top x \\ y &\in Y_{k-1,m}, \end{aligned}$$

which further leads to

$$\begin{aligned} \min \left\{ x^\top Aw : w \in Y_{k,m} \right\} + k\bar{a}^\top x - b &\geq x^\top Ay + (k-1)\bar{a}^\top x \\ y &\in Y_{k-1,m}. \end{aligned}$$

In particular, if $Q = [0, 1]^m$, from Theorem 1 we obtain the equivalent formulation

$$\begin{aligned} (\text{IP}_{\text{new}}) \quad \min \quad & c^\top x \\ \text{s.t.} \quad & g^\top \pi + \bar{a}^\top x \geq b + x^\top Ay \\ & E^\top \pi = A^\top x \\ & F^\top \pi \leq 0 \\ & y \in Y_{k-1,m} \\ & x \in X. \end{aligned}$$

for (P_v) . For general binary Q , our formulation reads

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & w^\top A^\top x + \bar{a}^\top x \geq b + x^\top Ay \quad \forall w \in Y_{k,Q} \\ & y \in Y_{k-1,Q} \\ & x \in X, \end{aligned}$$

for which we must resort on separation.

If $X \subseteq \{0, 1\}^n$, after linearizing $x^\top Ay$, we obtain an IP formulation for (P_v) without additional binary variables for each scenario in \mathcal{A} .

4 Extensions

4.1 Joint constraints

Consider the case where we have multiple constraints that must be satisfied at the same time. Whenever one or more of them are violated, we consider it as a failure. The problem to solve is

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & \left\{ a_q^\top x \geq b_q, 1 \leq q \leq r \right\} \quad \text{for all but at most } k-1 \text{ matrices } [a_1, \dots, a_r]^\top \in \mathcal{A} \\ & x \in X, \end{aligned}$$

or equivalently,

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & \min_{1 \leq q \leq r} \left\{ a_q^\top x - b_q \right\} \geq 0 \quad \text{for all but at most } k-1 \text{ matrices } [a_1, \dots, a_r]^\top \in \mathcal{A} \\ & x \in X. \end{aligned}$$

Following the approach presented in the previous section, we assume that $a_q = \bar{a}_q + A_q z$ for $z \in \text{vert}(Q)$. Note that the vector z is the same across all a_q , modeling possible dependencies. Now the problem reads

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & \min_{1 \leq q \leq r} \left\{ x^\top (\bar{a}_q + A_q z) - b_q \right\} \geq 0 \quad \text{for all but at most } k-1 \text{ vectors } z \in \text{vert}(Q) \\ & x \in X. \end{aligned}$$

Given $x \in X$, the problem $\min_{z \in Q} \min_{1 \leq q \leq r} \left\{ x^\top (\bar{a}_q + A_q z) - b_q \right\}$ has an optimal solution at a vertex of Q since $\varphi(x, z) := \min_{1 \leq q \leq r} \left\{ x^\top (\bar{a}_q + A_q z) - b_q \right\}$ is a concave function of z . Let $v_k(x)$ be the k -th smallest value of $\varphi(x, z)$ over $z \in \text{vert}(Q)$, $\sigma_k(x)$ the sum of the values of its k best basic solutions, and $\bar{\sigma}_k(x)$ their average.

Assuming $Q \subseteq \mathbb{R}^m$ is a binary polytope, and recalling $L_{k,Q} = L_{k,m} \cap Q^k$, we have

$$\begin{aligned} \bar{\sigma}_k(x) = \min \quad & \frac{1}{k} \sum_{i=1}^k \min_{1 \leq q \leq r} \left\{ x^\top A_q z_i + x^\top \bar{a}_q - b_q \right\} \\ \text{s.t.} \quad & z = [z_1, \dots, z_k]^\top \in L_{k,Q}. \end{aligned}$$

Therefore, we have that $\bar{\sigma}_k(x) \geq 0$ is equivalent to

$$\sum_{i=1}^k \min_{1 \leq q \leq r} \left\{ x^\top A_q z_i + x^\top \bar{a}_q - b_q \right\} \geq 0 \quad \forall [z_1, \dots, z_k]^\top \in L_{k,Q}. \quad (6)$$

Proposition 3. *Under Assumption 2, inequalities (6) can be separated in polynomial time.*

Proof. Let $x \in X$ and define $Z_0 := \emptyset$. For $i = 1, \dots, k$, let $Z_i := Z_{i-1} \cup \{z_i^*\} \subseteq \{0, 1\}^m$, where z_i^* is an optimal solution to

$$\min \left\{ \min_{1 \leq q \leq r} \left\{ x^\top A_q z_i + x^\top \bar{a}_q - b_q \right\} : z_i \in \text{vert}(Q) \setminus Z_{i-1} \right\}.$$

Since linear optimization over $\text{vert}(Q) \setminus Z_{i-1}$ can be done in polynomial time [2], we can find z_i^* efficiently by solving

$$\min \left\{ x^\top A_q z_i + x^\top \bar{a}_q - b_q : z_i \in \text{vert}(Q) \setminus Z_{i-1} \right\}$$

for each $1 \leq q \leq r$ and taking z_i^* as an optimal solution to the problem with the least objective value. Finally, order and relabel the elements in Z_k so that $z^* = [z_1^*, \dots, z_k^*]^\top \in L_{k,Q}$ and check whether x satisfies (6) for z^* or not. \square \square

In this setting, we have that $v_k(x) \geq 0$ is equivalent to $\sigma_k(x) \geq \sigma_{k-1}(x)$, which can be formulated as

$$\begin{aligned} \sum_{i=1}^k \min_{1 \leq q \leq r} \left\{ x^\top A_q \hat{z}_i + x^\top \bar{a}_q - b_q \right\} &\geq \theta \quad \forall \hat{z} = [\hat{z}_1, \dots, \hat{z}_k]^\top \in L_{k,Q} \\ \sum_{i=1}^{k-1} \min_{1 \leq q \leq r} \left\{ x^\top A_q z_i + x^\top \bar{a}_q - b_q \right\} &\leq \theta \\ z = [z_1, \dots, z_{k-1}]^\top &\in L_{k-1,Q} \\ \theta &\in \mathbb{R}, \end{aligned}$$

where we have introduced variable $\theta \in \mathbb{R}$ for simplicity. From Proposition 3, the first set of constraints is easy to separate. To represent the second set of constraints, we include additional variables $w \in \{0, 1\}^{(k-1) \times r}$ and $\eta \in \mathbb{R}^{k-1}$, yielding

$$\begin{aligned}
& \min && c^\top x \\
& \text{s.t.} && \sum_{i=1}^k \min_{1 \leq q \leq r} \left\{ x^\top A_q \hat{z}_i + x^\top \bar{a}_q - b_q \right\} \geq \theta \quad \forall \hat{z} = [\hat{z}_1, \dots, \hat{z}_k]^\top \in L_{k,Q} \\
& && \sum_{i=1}^{k-1} \eta_i \leq \theta \\
& && x^\top A_q z_i + x^\top \bar{a}_q - b_q \leq \eta_i + M_q w_{iq} \quad 1 \leq i < k, 1 \leq q \leq r \\
& && \sum_{q=1}^r w_{iq} \leq r - 1 \quad 1 \leq i < k \\
& && x \in X \\
& && z = [z_1, \dots, z_{k-1}]^\top \in L_{k-1,Q} \\
& && w \in \{0, 1\}^{k \times r} \\
& && \eta \in \mathbb{R}^k \\
& && \theta \in \mathbb{R}.
\end{aligned}$$

4.2 Integral parametrization

Now, suppose that \mathcal{A} is parametrized by the integral vectors within the rectangle $[0, u]^m$ for some $u \in \mathbb{Z}_+$. Again, finding the k best solutions to $\min\{x^\top(\bar{a} + Az) : z \in \{0, \dots, u\}^m\}$ can be done in polynomial time [2] and $\bar{\sigma}_k(x)$ can be computed as

$$\begin{aligned}
\bar{\sigma}_k(x) = & \min && \frac{1}{k} x^\top Ay + x^\top \bar{a} \\
& \text{s.t.} && \sum_{i=1}^k z_i = y \\
& && z_i \prec z_j \quad i < j \\
& && z_i \in \{0, \dots, u\}^m \quad 1 \leq i \leq k.
\end{aligned}$$

Let $L_{k,m}^u$ be the set of $k \times m$ integral matrices with rows in increasing lexicographical order and having entries in $\{0, \dots, u\}$. In [1], extended formulations for $\text{conv}(L_{k,m}^u)$ of size $\mathcal{O}(k^3m)$, if $u \geq k - 1$, and $\mathcal{O}(uk^3m)$, if $u < k - 1$, are obtained via dynamic programming.

Let $Y_{k,m}^u := \left\{ y \in \mathbb{Z}_+^m : y = \sum_{i=1}^k z_i \text{ for some } [z_1, \dots, z_k]^\top \in L_{k,m}^u \right\}$. Along the lines of Theorem 1, we can build an extended formulation for $\text{conv}(Y_{k,m}^u)$ of size $\mathcal{O}(k^2m)$ or $\mathcal{O}(uk^2m)$, depending on whether $u \geq k - 1$ or not, and then apply LP duality to obtain a compact representation of the constraint $\bar{\sigma}_k(x) \geq b$.

5 Computational experiments

We compare the standard IP formulation (IP_{std}) and our new formulation (IP_{new}) on problems of the form

$$\begin{aligned} \max \quad & \mathbb{E}[a]^\top x \\ \text{s.t.} \quad & a^\top x \geq b \text{ for all but at most } k - 1 \text{ vectors } a \in \mathcal{A} \\ & \mathbf{1}^\top x \leq n/2 \\ & x \in \{0, 1\}^n, \end{aligned}$$

where $\mathcal{A} \subseteq \mathbb{R}^n$ is as in Example 2, that is, we take $Q = [0, 1]^n$ and given $l \in \mathbb{R}^n$ and $g \in \mathbb{R}_+^n$, we set $a = l + \text{diag}(g - l)z$ for $z \in \{0, 1\}^n$. For each $n \in \{12, 14, 16, 18\}$, we generate 10 instances of g and l , where g_j and l_j are drawn uniformly from $\{1, \dots, 40\}$ and $\{-\lceil h_j/2 \rceil, \dots, -1\}$, respectively. For each instance, we have $\mathbb{E}[a] = (g + l)/2 \geq \mathbf{0}$ and we set $b = -0.2 \cdot \mathbf{1}^\top \mathbb{E}[a]$. Finally, for each instance, we formulate and solve the problem for $k \in \{10, 20, 30, 40, 50\}$ with (IP_{std}) and (IP_{new}).

Our implementation uses IBM CPLEX 12.8 as IP solver. The experiments were run single-threaded on a cluster with 15 nodes equipped with two Intel E5-2470 8-cores processors. In Tables 1–4 below we report the CPU time, in seconds, to reach optimality within the default solver tolerance with formulations (IP_{std}) and (IP_{new}).

Table 1: Time to optimality in seconds, $n = 12$.

Instance #	$k = 10$		$k = 20$		$k = 30$		$k = 40$		$k = 50$	
	Std	New	Std	New	Std	New	Std	New	Std	New
1	2.3	0.3	2.6	3.1	5.3	12.4	6.6	40.5	21.5	133.2
2	2.2	0.3	4.0	3.0	5.8	11.2	16.5	40.5	63.5	161.0
3	1.2	0.2	2.4	1.2	2.3	7.6	3.8	26.2	5.4	78.3
4	0.3	0.0	0.3	0.3	0.3	0.7	0.3	1.4	0.3	3.2
5	2.3	0.3	3.0	3.6	5.2	14.7	8.7	57.6	8.0	212.1
6	1.7	0.3	2.8	1.9	4.1	5.1	5.5	18.3	7.4	67.3
7	0.4	0.1	0.4	0.3	0.4	2.6	0.5	2.7	0.5	7.5
8	3.1	0.4	3.9	3.5	5.6	13.6	7.0	60.9	22.9	241.0
9	2.9	0.4	3.5	3.8	4.0	15.5	6.8	62.4	8.6	249.3
10	2.3	0.3	3.5	3.7	4.3	12.5	5.4	65.2	6.1	207.6
Average	1.9	0.3	2.6	2.4	3.7	9.6	6.1	37.6	14.4	136.1

Recall that (IP_{std}) has $\mathcal{O}(2^n)$ binary variables and constraints, since $|\mathcal{A}| = 2^n$, while (IP_{new}) has $\mathcal{O}(n)$ binary variables and $\mathcal{O}(k^2 n)$ continuous variables and constraints. We see that for both formulations, solving times increase with n and k , but at different rates. For (IP_{std}), the growth as a function of k is rather mild, while as a function of n it is much more pronounced. For (IP_{new}), the opposite happens, as the increase with k is much steeper than with n .

We also observe that for small k , (IP_{new}) outperforms (IP_{std}). For $n = 12$ and $n = 14$, both formulations become comparable for $k = 20$ and $k = 30$, respectively, while (IP_{std}) outperforms (IP_{new})

Table 2: Time to optimality in seconds, $n = 14$.

Instance #	$k = 10$		$k = 20$		$k = 30$		$k = 40$		$k = 50$	
	Std	New	Std	New	Std	New	Std	New	Std	New
1	17.2	0.4	22.9	5.1	27.5	24.1	34.6	153.3	37.5	398.9
2	7.1	0.2	6.1	1.6	5.8	7.4	7.8	11.0	8.0	46.9
3	7.7	0.4	11.7	5.8	18.3	25.8	22.0	138.7	36.9	378.4
4	15.0	0.4	20.6	4.6	29.1	21.6	45.8	71.7	53.0	258.3
5	10.3	0.5	17.0	6.3	21.0	26.5	36.3	119.5	57.2	425.9
6	12.9	0.3	14.9	2.9	15.7	10.5	22.2	49.6	34.0	131.8
7	15.1	0.3	18.7	6.9	30.9	33.9	36.2	151.7	33.1	321.4
8	12.2	0.4	16.0	5.1	19.7	18.6	25.4	102.1	31.4	253.1
9	12.4	0.3	15.6	5.7	22.1	26.3	20.4	128.4	27.0	345.5
10	15.2	0.5	22.6	6.8	44.7	28.9	69.2	145.5	123.9	391.4
Average	12.5	0.4	16.6	5.1	23.5	22.4	32.0	107.1	44.2	295.1

Table 3: Time to optimality in seconds, $n = 16$.

Instance #	$k = 10$		$k = 20$		$k = 30$		$k = 40$		$k = 50$	
	Std	New	Std	New	Std	New	Std	New	Std	New
1	94.6	0.3	121.2	3.3	146.5	12.8	189.6	77.7	240.6	252.1
2	48.9	0.2	54.9	2.1	66.6	13.7	69.9	39.2	73.6	148.2
3	111.6	0.7	130.4	4.3	159.7	28.8	222.0	127.6	281.1	415.5
4	48.6	0.3	62.7	2.7	63.9	8.5	78.7	43.5	89.3	102.6
5	136.0	0.5	177.9	7.4	232.7	36.9	277.2	234.3	392.3	530.2
6	103.7	0.5	145.5	8.3	179.7	32.3	215.4	153.1	302.6	440.5
7	87.3	0.4	127.9	7.5	168.0	28.3	322.7	122.1	277.5	524.9
8	74.8	0.7	106.0	10.2	124.5	65.5	181.2	282.9	214.2	873.7
9	132.2	0.9	160.8	9.6	202.1	32.9	281.2	184.1	353.5	635.3
10	133.1	0.8	187.4	8.9	242.7	38.7	327.0	260.9	444.8	841.9
Average	97.1	0.5	127.5	6.4	158.6	29.8	216.5	152.5	266.9	476.5

for larger k . For $n = 16$, the figure reverses only for $k = 50$, while for $n = 18$, (IP_{std}) is not able to compete with (IP_{new}) for any value of k .

In light of these results, we conclude that for problems of the form (P_{k-1}) with large $|\mathcal{A}|$ and small k , (IP_{new}) might preferable to (IP_{std}) .

Table 4: Time to optimality in seconds, $n = 18$.

Instance #	$k = 10$		$k = 20$		$k = 30$		$k = 40$		$k = 50$	
	Std	New	Std	New	Std	New	Std	New	Std	New
1	4319.2	0.8	4476.0	12.3	4686.1	88.3	4830.5	411.8	5004.5	1281.6
2	2635.0	0.4	2794.6	3.7	2961.4	12.9	3101.8	82.3	3371.4	166.8
3	470.3	0.4	568.3	4.6	670.3	22.6	735.2	63.5	820.2	261.9
4	295.0	0.5	358.8	2.3	428.4	8.6	503.6	33.3	586.4	91.3
5	2112.9	0.7	2455.2	12.8	2754.9	71.3	2766.4	529.5	3016.4	1705.6
6	1702.3	0.6	1797.3	8.5	1959.7	45.6	2052.8	340.7	2646.3	819.5
7	220.2	0.6	294.4	8.0	1973.2	45.4	422.7	290.5	2134.8	711.8
8	1200.9	0.3	1323.1	4.1	1418.3	50.8	1542.1	189.2	1614.2	383.2
9	1585.5	0.5	1715.4	9.6	1742.4	55.3	1989.4	531.0	2169.8	1371.9
10	520.7	0.6	638.2	5.8	795.3	37.3	1123.7	214.6	1574.7	565.8
Average	1506.2	0.5	1642.1	7.2	1939.0	43.8	1906.8	268.6	2293.9	735.9

6 Concluding remarks

In this work, we present novel formulations for a class of stochastic problems with exponentially many scenarios parametrized by the vertices of a polytope. This approach introduces a modest number of additional variables, and it is shown to outperform a standard IP formulation when the failure tolerance is small.

We want to highlight the fact that our compact formulations can be readily given to an off-the-shelf solver without the need of an intricate preprocessing or implementation. Formulations that include a large family of inequalities can be tackled with the built-in functions of state-of-the-art solvers. We hope that these characteristics broaden the applicability of the proposed formulations.

A natural question to further address is whether the approach can be extended to non-uniform discrete distributions, to either derive alternative formulations or to provide a fast mechanism to provide strong bounds. Also, objective functions or constraints that consider the expectation operator under similar settings might be considered too.

Acknowledgments

The author is grateful to Diego Morán for commenting on an earlier version of this work.

References

- [1] G. Angulo, *Matrices with lexicographically-ordered rows*, Optimization Letters **13** (2019), no. 2, 235–248.

- [2] G. Angulo, S. Ahmed, S.S. Dey, and V. Kaibel, *Forbidden vertices*, *Mathematics of Operations Research* **40** (2015), 35–360.
- [3] G.C. Calafiore and M.C. Campi, *Uncertain convex programs: randomized solutions and confidence levels*, *Mathematical Programming* **102** (2005), no. 1, 25–46.
- [4] ———, *The scenario approach to robust control design*, *Automatic Control, IEEE Transactions on* **51** (2006), no. 5, 742–753.
- [5] A. Charnes and W.W. Cooper, *Chance-constrained programming*, *Management science* **6** (1959), no. 1, 73–79.
- [6] U. Feige, K. Jain, M. Mahdian, and V. Mirrokni, *Robust Combinatorial Optimization with Exponential Scenarios*, *Integer Programming and Combinatorial Optimization* (M. Fischetti and D.P. Williamson, eds.), *Lecture Notes in Computer Science*, vol. 4513, Springer Berlin Heidelberg, 2007, pp. 439–453 (English).
- [7] Volker Kaibel and Andreas Loos, *Branched polyhedral systems*, *Integer programming and combinatorial optimization*, Springer, 2010, pp. 177–190.
- [8] Rohit Khandekar, Guy Kortsarz, Vahab Mirrokni, and Mohammad R Salavatipour, *Two-stage robust network design with exponential scenarios*, *Algorithmica* **65** (2013), no. 2, 391–408.
- [9] E.L. Lawler, *A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem*, *Management Science* **18** (1972), 401–405.
- [10] A. Loos, *Describing Orbitopes by Linear Inequalities and Projection Based Tools*, Ph.D. thesis, 2011.
- [11] J. Luedtke and S. Ahmed, *A sample approximation approach for optimization with probabilistic constraints*, *SIAM Journal on Optimization* **19** (2008), no. 2, 674–699.
- [12] R.K. Martin, R.L. Rardin, and B.A. Campbell, *Polyhedral characterization of discrete dynamic programming*, *Oper. Res.* **38** (1990), no. 1, 127–138.
- [13] B.L. Miller and H.M. Wagner, *Chance constrained programming with joint constraints*, *Operations Research* **13** (1965), no. 6, 930–945.
- [14] I. Muter, S.I. Birbil, and K. Bülbül, *Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows*, *Mathematical Programming* **142** (2013), no. 1-2, 47–82.
- [15] A. Nemirovski and A. Shapiro, *Convex approximations of chance constrained programs*, *SIAM Journal on Optimization* **17** (2006), no. 4, 969–996.
- [16] ———, *Scenario Approximations of Chance Constraints*, *Probabilistic and Randomized Methods for Design under Uncertainty* (G. Calafiore and F. Dabbene, eds.), Springer London, 2006, pp. 3–47 (English).
- [17] F. Qiu, S. Ahmed, S.S. Dey, and L.A. Wolsey, *Covering linear programming with violations*, *INFORMS Journal on Computing* **26** (2014), no. 3, 531–546.
- [18] T. Roos and P. Widmayer, *k-Violation linear programming*, *Information Processing Letters* **52** (1994), no. 2, 109–114.
- [19] Ruslan Sadykov and François Vanderbeck, *Column generation for extended formulations*, *EURO Journal on Computational Optimization* **1** (2013), no. 1-2, 81–115.