RAIRO-Oper. Res. 43 (2009) 157–187
DOI: 10.1051/ro/2009010

RAIRO Operations Research
www.rairo-ro.org

TRACTABLE ALGORITHMS FOR CHANCE-CONSTRAINED COMBINATORIAL PROBLEMS

Olivier Klopfenstein[1, 2]

Abstract. This paper aims at proposing tractable algorithms to find effectively good solutions to large size chance-constrained combinatorial problems. A new robust model is introduced to deal with uncertainty in mixed-integer linear problems. It is shown to be strongly related to chance-constrained programming when considering pure 0–1 problems. Furthermore, its tractability is highlighted. Then, an optimization algorithm is designed to provide possibly good solutions to chance-constrained combinatorial problems. This approach is numerically tested on knapsack and multi-dimensional knapsack problems. The results obtained outperform many methods based on earlier literature.

Keywords. Integer linear programming, chance constraints, robust optimization, heuristic.

Mathematics Subject Classification. 90C10, 90C15.

1. Introduction

Dealing with uncertainty in optimization problems has been a central topic since the fifty's [12]. Mainly because of the various natures of uncertainty in real-world issues, different frameworks of analysis have been proposed (for surveys, we refer to [23,25]). Among them, stochastic and robust approaches have to be

Received April 17, 2007. Accepted July 15, 2008.

[1] France Télécom R&D, 38-40 rue du gl Leclerc, 92130 Issy-les-Moulineaux, France; olivier.klopfenstein@orange-ftgroup.com

[2] Université de Technologie de Compiègne, Laboratoire Heudiasyc UMR CNRS 6599, 60205 Compiègne Cedex, France

Article published by EDP Sciences

distinguished. Stochastic programming relies on probabilistic information characterizing uncertainty, and it aims most of the time at optimizing the expected objective value. Chance-constrained programming appears as a specific model of stochastic optimization, where the goal is to optimize the objective, given an unfeasibility probability tolerance. The current paper focuses on this model. While this approach of uncertainty was introduced also in the fifty's [10], it is still considered as very difficult and widely intractable (see for instance [7,17]). For instance, the convexity of the set of solutions remains often an open question, even though some results clarify some specific cases. However, this stochastic model remains one of the most appropriate to deal with many decision problems, where unfeasibility is acceptable, provided that it occurs with a controlled probability.

By comparison with stochastic programming, robust optimization appears usually as simple and tractable. Its goal is to find the best solution which remains feasible for a whole set of possible events. The main criticism for robust models is the so-called over-conservatism: the obtained solution will be feasible for all the possible events, regardless their occurrence probability. In practice, the worst case may impose a large cost, while being highly improbable. To remedy this disadvantage, some previous works have proposed to relax this worst case condition [2,5]. The proposed robust models enable to obtain solutions which are known to remain feasible with high probability under quite weak probability assumptions (that is sometimes referred to as *ambiguous* chance constrained optimization, [13,21]). In these papers, the concept of robustness applies not to the entire set of events, but only to a subset of events. Hence, these robust approaches provide some feasible solutions to the corresponding chance-constrained problem. The central question is to know how good the robust solutions are, compared with an optimal solution to the chance-constrained problem. Several papers underlined explicitly the link between robust optimization and chance-constrained programming (*e.g.* [11,21]). The authors of [21] developed convex approximations of the set of chance-constrained feasible solutions. In particular, the obtained results, namely the so-called Bernstein approximation, improve the simpler framework of [2]. Furthermore, [18] tried to clarify the relations between the framework of [5] and chance-constrained programming for the specific case of the knapsack problem.

With respect to combinatorial chance-constrained problems, the literature is scarce. When considering a finite number of scenarios, a chance-constrained integer linear problem can be equivalently written as an integer linear program (see *e.g.* [15]). This equivalent model, while being natural, is highly intractable (see [24] for a work elaborating on this approach). Still with finite discrete scenarios, a branch-and-bound algorithm is proposed in [3] when only the right hand side of constraints is random, with joint probability constraints. [26] proposed a non-standard integer programming approach, based on Gröbner basis theory, to solve general chance-constrained integer programs to optimality. Unfortunately, without further improvements, the application of this method is restricted to small-size problems. Some other methods rely on sampling, and are readily usable with integer variables [8,13,22]. Their simplicity makes them very attractive at first sight. However, they may require a very large number of scenarios, and tend to provide too conservative solutions (see [21] for a comparison with the Bernstein approximation). Finally, metaheuristics have also been used to deal with chance-constrained problems, especially with integer variables (see *e.g.* [1,6]). The main practical difficulty in using metaheuristics lies in the tuning of parameters, which may require lots of tests.

Our aim in the current paper is to propose simple heuristic algorithms to find good solutions to general chance-constrained integer linear problems. A new robust optimization framework is introduced. It is shown to be strongly related to chance-constrained programming, while being highly tractable. In fact, this model enables a generalization of the results obtained in [18] for the specific case of knapsack problems. Relying on this robust model, an algorithm is designed to obtain satisfactory feasible solutions to chance-constrained integer linear problems. This solution method is applied on the knapsack problem and its multi-dimensional version. Numerical experiments illustrate the relevance and effectiveness of the proposed framework. Furthermore, it is simple to implement, highly tractable and scalable, and thus practically usable for large size real-life problems.

2. Problems and formulations

2.1. Pure integer linear programs

We consider the following integer linear problem (ILP):

$$\begin{array}{rc} \max & cx \\ \text{s.t.} & Ax \leq b \\ & x \in \{0,1\}^n \end{array} \tag{2.1}$$

where $A \in \mathbb{R}^{m\times n}$ is the constraint matrix, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. We denote $I = \{1,\ldots,n\}$ and $J = \{1,\ldots,m\}$. Suppose now that the matrix A is not known with certainty. More precisely, let us denote by $I_u(j) \subseteq I$ the set of uncertain coefficients for each row $j \in J$, any coefficient of $I \backslash I_u(j)$ being precisely known. Furthermore, we focus on the case where for all $j \in J$, for all $i \in I_u(j)$, A_{ji} lies in an interval $[\underline{A}_{ji}, \overline{A}_{ji}]$, with $\overline{A}_{ji} > \underline{A}_{ji}$. In the remaining, we will denote $\delta_{ji} = \overline{A}_{ji} - \underline{A}_{ji}$. When $i \notin I_u(j)$, the notation $\underline{A}_{ji}$ will sometimes be used instead of A_{ji} to simplify notations.

Note that considering that only the coefficients of A can be uncertain is not restrictive: this setting includes the cases where cost coefficients or right hand side coefficients are uncertain. Indeed, when b is uncertain, it is sufficient to add a dummy variable $x_{n+1} = 1$ and to write: $Ax - bx_{n+1} \leq 0$. When the objective vector c is subject to uncertainty, we may add a new variable $z \in \mathbb{R}$ and consider the problem: $\max_{x,z} \left\{z \mid z \leq cx, Ax \leq b, x \in \{0,1\}^n, z \in \mathbb{R}\right\}$. Even though a continuous variable is added, all the analysis of the current paper remains applicable (see Sect. 2.2).

Let $\mathcal{A}$ denote the set of constraint matrices A satisfying the above description.

Each uncertain coefficient is associated with a random variable, also denoted by A_{ji}, taking values in $[\underline{A}_{ji}, \overline{A}_{ji}]$. We also introduce: $\eta_{ji} = (A_{ji} - \underline{A}_{ji})/\delta_{ji}$, which is a random variable taking values in $[0, 1]$. Then, let us introduce the associated chance-constrained problem (CCILP):

$$\begin{array}{rc} \max & cx \\ \text{s.t.} & P(Ax \leq b) \geq 1 - \varepsilon, \\ & x \in \{0,1\}^n. \end{array} \tag{2.2}$$

P denotes the probability measure, and $\varepsilon \in [0, 1)$ is the unfeasibility tolerance for a given solution. In other words, solving CCILP means finding the best solution x feasible with probability at least $1 - \varepsilon$.

CCILP is the main problem we want to tackle. However, since chance-constrained programming is known to be very hard, our aim is to use a simpler parameterized robust model to approximate it. That is the reason for introducing the robust problem RILP(γ), parameterized by a vector $\gamma \in [0, 1]^m$:

$$\begin{array}{rc} \max & cx \\ \text{s.t.} & \sum_{i \in I} \underline{A}_{ji} x_i + \Delta_j(x) \leq b_j, \quad \forall j \in J \\ & x \in \{0,1\}^n \end{array} \tag{2.3}$$

with:

$$\Delta_j(x) = \min \left\{ \sum_{i \in I_u(j)} \delta_{ji} x_i,\ \gamma_j . \sum_{i \in I_u(j)} \delta_{ji} \right\}.$$

RILP(γ) will often be simply denoted RILP. $\Delta_j(x)$ is the protection term of the robust constraint $j \in J$. Observe that if $\gamma_j = 0$, the protection term is zero, that implies that only the best scenario is taken into account for row j. On the contrary, $\gamma_j = 1$ means that the worst case is taken into account, when all the coefficients of row j can take their largest values.

2.2. Mixed integer linear programs

For the sake of simplicity, all the paper is written in a pure integer linear context. However, all the results are readily available for mixed integer linear problems of the following form:

$$\begin{array}{rcl} \max & cx & \\ \text{s.t.} & Ax \leq b & \\ & x_i \in \{0,1\}, & \forall i \in I_1 \\ & x_i \geq 0, & \forall i \in I_2 \end{array}$$

when considering that for each row j, the uncertain coefficients only impact integral components of x, *i.e.*: $I_u(j) \subseteq I_1$.

If $I_u(j) \not\subseteq I_1$, the robust model proposed may also be used. In this case, several results of the paper would remain correct, in particular those related to probability and complexity analysis. Some others can be easily, but carefully, adapted by considering that continuous variables lie in $[0,1]$ (see for instance the reformulation of Sect. 4.2). However, the relation to chance-constrained programming becomes unclear, and the algorithms designed cannot be used.

2.3. On separate probability constraints

So far, only *joint* chance constraints have been considered (*cf.* model (2.2)). But some applications may require to consider *separate* chance constraints:

$$\begin{array}{lcl} \max & cx & \\ \text{s.t.} & P(A_j x \le b_j) \ge 1 - \varepsilon_j, & \forall j \in J \\ & x_i \in \{0,1\}, & \forall i \in I \end{array} \tag{2.4}$$

where $\varepsilon_j \in [0,1)$. For the sake of simplicity, all the results will be stated and proved for joint probability constraints. However, they can all be easily transposed to the case of separate probability constraints.

3. On the link between RILP and CCILP

Let $I' \subseteq I$ and $\gamma \in [0,1]^m$. We introduce the problem RILP(I',γ):

$$\begin{array}{ll} \max & cx \\ \text{s.t.} & \sum_{i \in I_u(j) \cap I'} \underline{A}_{ji} x_i + \Delta_j(I', x) + \sum_{i \in I_u(j) \setminus I'} \overline{A}_{ji} x_i \\ & \qquad + \sum_{i \notin I_u(j)} A_{ji} x_i \le b_j, \quad \forall j \in J \\ & x \in \{0,1\}^n \end{array} \tag{3.1}$$

where: $\Delta_j(I', x) = \min \left\{ \sum_{i \in I_u(j) \cap I'} \delta_{ji} x_i, \gamma_j . \sum_{i \in I_u(j) \cap I'} \delta_{ji} \right\}$. When considering the problem RILP(I',γ), all uncertain coefficients $\{A_{ji}\}_{j \in J, i \notin I'}$ are assumed to take their maximal values. Hence, for a row $j \in J$, the robustness parameter γ_j impacts only coefficients $\{A_{ji}\}_{i \in I'}$. Note that in this setting, RILP(γ) is equivalently denoted by RILP(I,γ).

Theorem 3.1. *There exists $I^* \subseteq I$ and $\gamma^* \in [0,1]^m$ such that any optimal solution of RILP(I^*,γ^*) is an optimal solution of CCILP.*

Proof. Let x^* be an optimal solution of CCILP. Let us introduce: $I^* = \{i \in I | x^*_i = 1\}$; to simplify notations, we also denote: $I^*(j) = I^* \cap I_u(j)$ for all $j \in J$. x^* is feasible with probability at least $1 - \varepsilon$, *i.e.*: $P(Ax^* \le b) \ge 1 - \varepsilon$. Let us denote $\mathcal{A}^* = \{A \in \mathcal{A} | Ax^* \le b\}$, and consider the following problem:

$$\begin{array}{lcl} \max & cx & \\ \text{s.t.} & Ax \le b, & \forall A \in \mathcal{A}^* \\ & x \in \{0,1\}^n. & \end{array} \tag{3.2}$$

Let us prove first that any optimal solution x' of (3.2) is also optimal for CCILP. We have: $P(Ax' \le b) \ge P(A \in \mathcal{A}^*) = P(Ax^* \le b) \ge 1 - \varepsilon$. Thus, x' is a feasible solution of CCILP, that implies: $cx^* \geq cx'$. But as x^* is feasible for (3.2), we also have: $cx^* \leq cx'$. Hence, x' is a feasible solution of CCILP with the same cost than x^*: x' is optimal for CCILP.

The goal now is to find parameters $(\gamma_j^*)_{j\in J}$ such that (3.2) is equivalent to RILP(I^*,γ^*). Let $j \in J$, we have: $A_j x^* \leq b_j \Leftrightarrow \sum_{i\in I_u(j)} A_{ji}x_i^* \leq b_j - \sum_{i\notin I_u(j)} A_{ji}x_i^* \Leftrightarrow \sum_{i\in I_u(j)} \delta_{ji}\eta_{ji}x_i^* \leq b_j - \sum_{i\notin I_u(j)} A_{ji}x_i^* - \sum_{i\in I_u(j)} \underline{A}_{ji}x_i^*$. Let us denote: $\gamma_j^* = \min\left\{1, \left(b_j - \sum_{i\notin I_u(j)} A_{ji}x_i^* - \sum_{i\in I_u(j)} \underline{A}_{ji}x_i^*\right)/\sum_{i\in I_u(j)} \delta_{ji}\right\}$. Note that since $\mathcal{A}^* \neq \phi$, $\gamma_j^* \geq 0$. Then: (a) $A \in \mathcal{A}^* \Leftrightarrow \forall j \in J, \sum_{i\in I^*(j)} \delta_{ji}\eta_{ji} \leq \gamma_j^* \sum_{i\in I^*(j)} \delta_{ji}$.

Let us prove that x is a feasible solution of (3.2) if and only if it is feasible for RILP(I^*,γ^*). Let x be a feasible solution of (3.2). Let $j \in J$. If $\sum_{i\in I^*(j)} \delta_{ji}x_i \leq \gamma_j^* \sum_{i\in I^*(j)} \delta_{ji}$, let A_j be defined by: $\eta_{ji} = x_i$ if $i \in I^*(j)$, and $\eta_{ji} = 1$ if $i \in I_u(j)\backslash I^*$. We have: $\sum_{i\in I^*(j)} \delta_{ji}\eta_{ji} = \sum_{i\in I^*(j)} \delta_{ji}x_i \leq \gamma_j^* \sum_{i\in I^*(j)} \delta_{ji}$. According to (a), this ensures that $A \in \mathcal{A}^*$.

If on the contrary $\sum_{i\in I^*(j)} \delta_{ji}x_i > \gamma_j^* \sum_{i\in I^*(j)} \delta_{ji}$, let A_j be defined by:

$$\begin{array}{lcl} \text{if } i \in I^*(j) & : & \eta_{ji} = x_i.\gamma_j^*.\sum_{k\in I^*(j)} \delta_{jk}/\sum_{k\in I^*(j)} \delta_{jk}x_k \\ \text{if } i \in I_u(j)\backslash I^* & : & \eta_{ji} = 1. \end{array}$$

For the $\{\eta_{ji}\}_{i\in I_u(j)}$ to be admissible, we have to check that they are not larger than 1. This is clear when $i \in I_u(j)\backslash I^*$. When $i \in I^*(j)$, since $\sum_{i\in I^*(j)} \delta_{ji}x_i > \gamma_j^* \sum_{i\in I^*(j)} \delta_{ji}$, we have $\eta_{ji} \leq x_i$, and thus $\eta_{ji} \leq 1$. Moreover:

$$\begin{array}{rcl} \sum_{i\in I^*(j)} \delta_{ji}\eta_{ji} & = & \sum_{i\in I^*(j)} \delta_{ji}x_i.\gamma_j^*.\sum_{k\in I^*(j)} \delta_{jk}/\sum_{k\in I^*(j)} \delta_{jk}x_k \\ & = & \gamma_j^*.\sum_{k\in I^*(j)} \delta_{jk}. \end{array}$$

Thus, it is shown that $A \in \mathcal{A}^*$.

So far, A has been built in each case so that it belongs to $\mathcal{A}^*$. Moreover, for $j \in J$, let us show that: $A_j x = \sum_{i\in I^*(j)} \underline{A}_{ji}x_i + \Delta_j(I^*,x) + \sum_{i\in I_u(j)\backslash I^*} \overline{A}_{ji}x_i + \sum_{i\notin I_u(j)} A_{ji}x_i$. In the first case ($\sum_{i\in I^*(j)} \delta_{ji}x_i \leq \gamma_j^* \sum_{i\in I^*(j)} \delta_{ji}$): $\Delta_j(I^*,x) = \sum_{i\in I^*(j)} \delta_{ji}x_i$. As a consequence, from the definition of η_j in this case: $A_j x = \sum_{i\in I_u(j)}(\underline{A}_{ji} + \eta_{ji}\delta_{ji})x_i + \sum_{i\notin I_u(j)} A_{ji}x_i = \sum_{i\in I^*(j)} \underline{A}_{ji}x_i + \Delta_j(I^*,x) + \sum_{i\in I_u(j)\backslash I^*} \overline{A}_{ji}x_i + \sum_{i\notin I_u(j)} A_{ji}x_i$ (observing that $x_i^2 = x_i$). Let us now consider the second case ($\sum_{i\in I^*(j)} \delta_{ji}x_i > \gamma_j^* \sum_{i\in I^*(j)} \delta_{ji}$): $\Delta_j(I^*,x) = \gamma_j^* \sum_{i\in I^*(j)} \delta_{ji}$. Hence: $A_j x = \sum_{i\in I_u(j)}(\underline{A}_{ji} + \eta_{ji}\delta_{ji})x_i + \sum_{i\notin I_u(j)} A_{ji}x_i = \sum_{i\in I^*(j)} \underline{A}_{ji}x_i + \Delta_j(I^*,x) + \sum_{i\in I_u(j)\backslash I^*} \overline{A}_{ji}x_i + \sum_{i\notin I_u(j)} A_{ji}x_i$.

Thus, in both cases: $A_j x = \sum_{i\in I^*(j)} \underline{A}_{ji}x_i + \Delta_j(I^*,x) + \sum_{i\in I_u(j)\backslash I^*} \overline{A}_{ji}x_i + \sum_{i\notin I_u(j)} A_{ji}x_i$. As x is feasible for (3.2), we have: $A_j x \leq b_j$, which implies that x is feasible also for RILP(I^*,γ^*).

Suppose now that x is a feasible solution of RILP(I^*,γ^*). Let any $A \in \mathcal{A}^*$, we have for $j \in J$: $A_j x = \sum_{i\in I_u(j)}(\underline{A}_{ji} + \delta_{ji}\eta_{ji})x_i + \sum_{i\notin I_u(j)} A_{ji}x_i \leq \sum_{i\in I^*(j)} \underline{A}_{ji}x_i +$

$\sum_{i\in I^*(j)} \delta_{ji}\eta_{ji}x_i + \sum_{i\in I_u(j)\setminus I^*} \overline{A}_{ji}x_i + \sum_{i\notin I_u(j)} A_{ji}x_i$. Since $A \in \mathcal{A}^*$, the statement: $\sum_{i\in I^*(j)} \delta_{ji}\eta_{ji} \leq \gamma_j^* \sum_{i\in I^*(j)} \delta_{ji}$ holds. This implies, since $x_i \leq 1$ for all $i \in I^*(j)$: $\sum_{i\in I^*(j)} \delta_{ji}\eta_{ji}x_i \leq \gamma_j^* \sum_{i\in I^*(j)} \delta_{ji}$. On the other hand, since $\eta_{ji} \leq 1$ for all $i \in I^*(j)$: $\sum_{i\in I^*(j)} \delta_{ji}\eta_{ji}x_i \leq \sum_{i\in I^*(j)} \delta_{ji}x_i$. As a result: $\sum_{i\in I^*(j)} \delta_{ji}\eta_{ji}x_i \leq \Delta_j(I^*, x)$. Thus, we obtain: $A_jx \leq \sum_{i\in I^*(j)} \underline{A}_{ji}x_i + \Delta_j(I^*, x) + \sum_{i\in I_u(j)\setminus I^*} \overline{A}_{ji}x_i + \sum_{i\notin I_u(j)} A_{ji}x_i$. As x is feasible for RILP(I^*,γ^*), we obtain: $A_jx \leq b_j$. Thus, it is proved that x is feasible for (3.2).

Finally, we have that x is a feasible solution of (3.2) if and only if it is feasible for RILP(I^*,γ^*). As both problems have the same objective function, they are completely equivalent. As a consequence, any optimal solution of RILP(I^*,γ^*) is optimal also for CCILP. □

This establishes the theoretical link between CCILP and RILP. Note in particular that this result does not require any probability assumption.

4. Tractability of RILP

4.1. Theoretical complexity

That RILP is NP-hard if ILP is NP-hard is immediate, since ILP is a special case of RILP (consider $\gamma_j = 0$ for all $j \in J$). More precise complexity connections can be established between ILP and RILP. Hereafter, m' denotes the number of constraints $j \in J$ such that $I_u(j) \neq \phi$.

Proposition 4.1. *An optimal solution of RILP can be obtained by solving $2^{m'}$ problems ILP.*

Proof. In this proof, for the sake of simplicity, it is assumed that all sets $I_u(j)$ are non-empty; as a result, we have here: $m' = m$. Let V be the optimal value of RILP. Let $\alpha \in \{0,1\}^{m'}$, let $j \in J$, we denote within this proof: $\Delta_j(x, \alpha_j) = (1 - \alpha_j).\left(\sum_{i\in I_u(j)} \delta_{ji}x_i\right) + \alpha_j.\left(\gamma_j \sum_{i\in I_u(j)} \delta_{ji}\right)$. Then, let us introduce:

$$\begin{array}{llcr} V(\alpha) = & \max & cx & \\ & \text{s.t.} & \sum_{i\in I_u(j)} \underline{A}_{ji}x_i + \Delta_j(x, \alpha_j) + \sum_{i\notin I_u(j)} A_{ji}x_i \leq b_j, & \forall j \in J \\ & & x_i \in \{0,1\}, & \forall i \in I. \end{array}$$

For any $\alpha \in \{0,1\}^m$, a feasible solution of the above problem is feasible also for RILP, since $\Delta_j(x, \alpha_j) \geq \Delta_j(x)$. It follows that: $\max\{V(\alpha) \mid \alpha \in \{0,1\}^m\} \leq V$. Furthermore, there exists necessarily $\alpha^* \in \{0,1\}^m$ such that $V \leq V(\alpha^*)$. Indeed, given an optimal solution x^* of RILP, it is sufficient to consider: $\alpha_j^* = 1 \Leftrightarrow \sum_{i\in I_u(j)} \delta_{ji}x_i^* \geq \gamma_j \sum_{i\in I_u(j)} \delta_{ji}$. In this case, x^* is a feasible solution of the corresponding problem.

Thus, it is shown that $V = \max\{V(\alpha) \mid \alpha \in \{0,1\}^m\}$. □

Thus, for instance, if the number m' of constraints subject to uncertainty does not impact the complexity of ILP, *i.e.* any instance of ILP has a fixed number of uncertain constraints, its robust version RILP can be solved through a constant number of problems ILP. For instance, this is the case for knapsack problems, where the classical formulation presents only one constraint. This result brings a similar consequence for the case where only cost coefficients are assumed to be uncertain, which can be solved through only 2 nominal problems.

The above proposition leads to the following observation:

Corollary 4.1. *Suppose that any instance of ILP has $O(\log(n))$ constraints. If ILP can be solved in polynomial time, this is true also for RILP.*

However, most of the time, the complexity of ILP also involves the number m of constraints, which may not be $O(\log(n))$. In this case, Proposition 4.1 leads to solve an exponential number of ILP instances. Nevertheless, in some specific cases, this can be a lot improved:

Proposition 4.2. *Suppose that for all $j \in J$, $I_u(j) = I_u \subseteq I$ and: $\forall i \in I_u, \delta_{ji} = \delta_j$. Then an optimal solution of RILP can be obtained by solving $m' + 1$ problems ILP.*

Proof. We continue the previous proof, where it has been shown that: $V = \max\{V(\alpha) | \alpha \in \{0,1\}^m\}$. As before, we denote by x^* an optimal solution of RILP, and $\alpha^* \in \{0,1\}^m$ such that: $\alpha^*_j = 1 \Leftrightarrow \sum_{i \in I_u(j)} \delta_{ji} x^*_i \geq \gamma_j \sum_{i \in I_u(j)} \delta_{ji}$. We know that $V = V(\alpha^*)$. Let us assume now that for all $j \in J$, $I_u(j) = I_u$ and all variations satisfy: $\delta_{ji} = \delta_j$. We sort the indices so that: $j < k \Rightarrow \gamma_j \leq \gamma_k$. Let any $j \in J$. Consider the case where $\sum_{i \in I_u} \delta_j x^*_i < \gamma_j \sum_{i \in I_u} \delta_j$, *i.e.*: $\alpha^*_j = 0$. This implies that for all $k \geq j$: $\sum_{i \in I_u} x^*_i < \gamma_k |I_u|$, and thus: $\sum_{i \in I_u} \delta_k x^*_i < \gamma_k \sum_{i \in I_u} \delta_k$. As a result, for all $k \geq j$: $\alpha^*_k = 0$. Thus, we have: $\alpha^*_j = 0 \Rightarrow \forall k \geq j, \alpha^*_k = 0$.

This implies that: $V = \max\big\{V(\alpha) | \alpha \in \{0,1\}^m, j \leq k \Rightarrow \alpha_j \geq \alpha_k\big\}$. As a consequence, an optimal solution of RILP can be obtained by solving $m + 1$ problems ILP, and keeping the best solution encountered. □

In this latter case, if ILP is solved in polynomial time, this remains true also for RILP. Note that Proposition 4.2 applies in particular when only right hand side data of a mixed integer linear program are subject to uncertainty.

In the case of "proportional variations", RILP is related to a simpler robust model, equivalent to ILP:

Proposition 4.3. *Suppose that for all $j \in J$, $I_u(j) = I$ and there exists $\kappa_j > 0$ such that: $\forall i \in I, \delta_{ji} = \kappa_j \underline{A}_{ji}$. Let $\gamma \in [0,1]^m$, there exists $\gamma' \in [0,1]^m$ such that any optimal solution of the problem (4.1) below is optimal also for RILP(γ):*

$$\begin{array}{lll} \max & cx & \\ s.t. & \sum_{i \in I} \underline{A}_{ji} x_i + \gamma'_j . \sum_{i \in I} \delta_{ji} \leq b_j, & \forall j \in J, \\ & x \in \{0,1\}^n & \forall i \in I. \end{array} \tag{4.1}$$

Proof. Let x^* be an optimal solution of RILP(γ). We define γ' through, for all $j \in J$: $\gamma'_j = \big(b_j - \sum_{i \in I} \underline{A}_{ji} x^*_i\big) / \sum_{i \in I} \delta_{ji}$. Observe that: $\gamma'_j \sum_{i \in I} \delta_{ji} \geq \Delta_j(x^*)$.

This implies that, when $\Delta_j(x^*) = \gamma_j.\sum_{i\in I}\delta_{ji}$, we have $\gamma'_j \geq \gamma_j$. As a result, $\gamma'_j < \gamma_j$ only if $\Delta_j(x^*) = \sum_{i\in I}\delta_{ji}x^*_i$.

Let x' be an optimal solution of (4.1). By construction, x^* is a feasible solution of (4.1), and then: $cx^* \leq cx'$. On the other hand, let us prove by contradiction that x' is feasible for RILP(γ). Suppose that this is not the case: there exists $j \in J$ such that $\Delta_j(x') > \gamma'_j.\sum_{i\in I}\delta_{ji}$. It is equivalent to say: $\sum_{i\in I}\delta_{ji}x'_i > \gamma'_j.\sum_{i\in I}\delta_{ji}$ and $\gamma_j\sum_{i\in I}\delta_{ji} > \gamma'_j.\sum_{i\in I}\delta_{ji}$. The latter condition means that $\gamma'_j < \gamma_j$, that occurs only when $\Delta_j(x^*) = \sum_{i\in I}\delta_{ji}x^*_i$. By definition of γ'_j, we have also: $\Delta_j(x^*) \leq \gamma'_j.\sum_{i\in I}\delta_{ji}$. Then, we deduce: $\sum_{i\in I}\delta_{ji}x'_i > \sum_{i\in I}\delta_{ji}x^*_i$.

As we assumed that $\delta_j = \kappa_j\underline{A}_j$, we obtain: $\sum_{i\in I}\underline{A}_{ji}x'_i > \sum_{i\in I}\underline{A}_{ji}x^*_i$. But, as x' is feasible for (4.1): $\sum_{i\in I}\underline{A}_{ji}x'_i \leq b_j - \gamma'_j.\sum_{i\in I}\delta_{ji}$. Furthermore, by definition of γ'_j: $\sum_{i\in I}\underline{A}_{ji}x^*_i = b_j - \gamma'_j\sum_{i\in I}\delta_{ji}$. This is a contradiction.

Hence, it is proved that x' is feasible for RILP(γ). This ensures that x' is an optimal solution of RILP(γ). □

The parameter γ' may not be directly deduced from γ. However, this result is of importance in the context of the iterative approaches detailed in Section 5. Indeed, under the assumptions of Proposition 4.3, the robust model RILP can be replaced by the simpler robust model (4.1).

4.2. An integer linear formulation of RILP

The continuous relaxation of RILP as written in (2.3) is not linear, it is even not convex since:

Lemma 4.1. *Let $j \in J$, $x \longmapsto \Delta_j(x)$ is concave on $[0,1]^n$.*

Proof. Suppose that $x = \lambda x^1 + (1-\lambda)x^2$ for some $\lambda \in [0,1]$, we have: $\sum_{i\in I_u(j)} x_i = \lambda\sum_{i\in I_u(j)} x^1_i + (1-\lambda)\sum_{i\in I_u(j)} x^2_i$. Then:

$$\min\left\{\sum_{i\in I_u(j)}\delta_{ji}x_i, \gamma_j.\sum_{i\in I_u(j)}\delta_{ji}\right\} = \min\left\{\lambda\gamma_j.\sum_{i\in I_u(j)}\delta_{ji} + (1-\lambda)\gamma_j.\sum_{i\in I_u(j)}\delta_{ji},\right.$$
$$\left.\lambda\sum_{i\in I_u(j)}\delta_{ji}x^1_i + (1-\lambda)\sum_{i\in I_u(j)}\delta_{ji}x^2_i\right\} \geq \lambda\min\left\{\gamma_j.\sum_{i\in I_u(j)}\delta_{ji}, \sum_{i\in I_u(j)}\delta_{ji}x^1_i\right\}$$
$$+ (1-\lambda)\min\left\{\gamma_j.\sum_{i\in I_u(j)}\delta_{ji}, \sum_{i\in I_u(j)}\delta_{ji}x^2_i\right\}.$$

This shows that: $\Delta_j(x) \geq \lambda\cdot\Delta_j(x^1) + (1-\lambda)\cdot\Delta_j(x^2)$. □

However, we have this equivalent formulation of RILP:

Proposition 4.4. *RILP can be equivalently written:*

$$\begin{array}{lll} \max & cx & \\ s.t. & \sum_{i\in I_u(j)} \underline{A}_{ji}x_i + \left(\gamma_j . \sum_{i\in I_u(j)} \delta_{ji}\right) .z_j & \\ & \quad + \sum_{i\in I_u(j)} \delta_{ji}y_{ji} + \sum_{i\notin I_u(j)} A_{ji}x_i \leq b_j, & \forall j \in J \\ & z_j + y_{ji} \geq x_i, & \forall j \in J, i \in I_u(j) \\ & x_i \in \{0,1\}, & \forall i \in I \\ & z_j \geq 0, y_{ji} \geq 0, & \forall j \in J, i \in I_u(j) \end{array} \qquad (4.2)$$

Proof. For any $x \in \{0,1\}^n$, it is easy to see that for all $j \in J$:

$$\Delta_j(x) = \min\left\{\left(\gamma_j \cdot \sum_{i\in I_u(j)} \delta_{ji}\right) \cdot z_j + \left(\sum_{i\in I_u(j)} \delta_{ji}x_i\right) \cdot (1-z_j) | z_j \in [0,1]\right\}.$$

This can be classically linearized by introducing $y_{ji} = x_i.(1-z_j) = \max\{0, x_i - z_j\}$:

$$\begin{array}{rclll} \Delta_j(x) & = & \min & \left(\gamma_j \cdot \sum_{i\in I_u(j)} \delta_{ji}\right) \cdot z_j + \sum_{i\in I_u(j)} \delta_{ji}y_{ji} & \\ & & \text{s.t.} & z_j + y_{ji} \geq x_i, & \forall i \in I_u(j) \\ & & & y_{ji} \geq 0, & \forall i \in I_u(j) \\ & & & z_j \in [0,1]. & \end{array}$$

Finally, the constraints $z_j \leq 1$ can be omitted. Indeed, any optimal solution (x, y, z) of (4.2) can be changed into (x, y', z') satisfying $z'_j + y'_{ji} = x_i$, for all $j \in J$ and $i \in I_u(j)$. As $x_i \leq 1$ for $i \in I$ and $y \geq 0$, this implies that $z'_j \leq 1$ for all $j \in J$. Thus, constraints $z_j \leq 1$ are unnecessary.

Hence, from any solution of RILP, one can build a solution of (4.2), and reciprocally. □

Hence, relaxing the integrality constraints of (4.2), we obtain a fully linear relaxation of RILP; we denote it by RILP-L.

Remark 4.1. The integrality constraints on z can be relaxed in (4.2) because x is a $\{0,1\}$-vector. But if the problem considered contains non-integral variables $x_i \in [0,1]$ associated to some uncertain coefficient A_{ji}, the exact reformulation of the robust problem requires to keep the constraint $z_j \in \{0,1\}$.

The formulation (4.2) can readily be compared with the robust model introduced in [5], which is:

$$\begin{array}{lll} \max & cx & \\ \text{s.t.} & \sum_{i\in I_u(j)} \underline{A}_{ji}x_i + \Gamma_j z_j + \sum_{i\in I_u(j)} \delta_{ji}y_{ji} & \\ & \quad + \sum_{i\notin I_u(j)} A_{ji}x_i \leq b_j, & \forall j \in J \\ & z_j + \delta_{ji}y_{ji} \geq \delta_{ji}x_i, & \forall j \in J, i \in I_u(j) \\ & x_i \in \{0,1\}, & \forall i \in I \\ & z_j \geq 0, y_{ji} \geq 0, & \forall j \in J, i \in I_u(j) \end{array} \qquad (4.3)$$

with robustness parameters $\Gamma_j \in [0, |I_u(j)|]$ for $j \in J$. In particular, observe that if all the coefficient variations δ_{ji} are equal to a same value δ, then (4.3) is completely equivalent to RILP by considering $\Gamma_j = \gamma_j |I_u(j)|$. That is the reason for the results of [18], where the approach of [5] was proved to be strongly related to chance-constrained programming for the uncertain knapsack problem when all coefficients variations were identical.

The robust model (4.3) will be denoted by $\text{RILP}_2(\Gamma)$. The model RILP, related to RILP_2, has been proposed because of its lower theoretical complexity (compare the results given in Section 4.1 with those in [4]), and because of its very easy approximability (see the "alternative fast algorithm" in Sect. 5.3).

5. Obtaining good solutions to CCILP: algorithmic aspects

5.1. Computing the feasibility probability of a solution

One of the key issues in solving optimization problems under probabilistic constraints is to assess the feasibility probability $P(Ax \leq b)$ of a given point x. To calculate exactly this would require complicated multivariate numerical integrations, that appears far too hard to design a tractable resolution process. This motivates the approximation of $P(Ax \leq b)$, which can be performed through several classical ways. We present hereafter three different approaches, depending on the probability assumptions made.

Probability bounds. If the probability distributions of uncertain coefficients are unknown, we can rely on some Hoeffding-type bound [16]:

Theorem 5.1 (Hoeffding 1963)**.** *Let X_1,...,X_n be independent random variables such that, for all $i \in \{1,\dots,n\}$: $P(X_i \in [\alpha_i, \beta_i]) = 1$. Then, denoting $S = \sum_{i=1}^n X_i$:*

$$\forall \tau \geq 0, P(S \geq E[S] + \tau) \leq \exp\left(-\frac{2\tau^2}{\sum_{i=1}^n (\beta_i - \alpha_i)^2}\right).$$

This result can be applied in our context, by observing that: $P(A_j x \geq b_j) = P\big(A_j x \geq E[A_j x] + (b_j - E[A_j x])\big)$. Then, with $S = A_j x$ and $\tau = b_j - E[A_j x]$, the following lemma comes:

Lemma 5.1. *Consider any $x \in I\!R^n$. Let $j \in J$, we denote: $\mu_j(x) = E\left[\sum_{i \in I_u(j)} A_{ji} x_i\right]$. Suppose that: $b_j \geq \mu_j(x) + \sum_{i \notin I_u(j)} A_{ji} x_i$, and that the random variables $\{A_{ji}\}_{i \in I_u(j)}$ are independent:*

$$P(A_j x \geq b_j) \leq \exp\left(-\frac{2 d_j(x)^2}{\sum_{i \in I_u(j)} \delta_{ji}^2 x_i^2}\right) \tag{5.1}$$

where: $d_j(x) = b_j - \sum_{i \notin I_u(j)} A_{ji} x_i - \mu_j(x)$.

When considering joint probability constraints, the global probability must be assessed: $P(Ax \leq b) = P(\forall j \in J, A_j x \leq b_j)$. For instance, if independence is assumed between random variables associated to uncertain coefficients of different constraints, we have: $P(Ax \leq b) = \prod_{j \in J} P(A_j x \leq b_j)$.

Note that the use of such analytical bounds $B(x) \leq P(Ax \leq b)$ means that the chance-constrained problem CCILP is in fact approximated through the following integer program:

$$\begin{array}{rl} \max & cx \\ \text{s.t.} & B(x) \geq 1 - \varepsilon \\ & x_i \in \{0,1\}, \quad \forall i \in I. \end{array} \tag{5.2}$$

This problem is in general non-linear, and thus often intractable. But in some special cases, it may be linearized. This is the case, for instance, when considering the bound (5.1) and separate probability constraints (*cf.* model (2.4)). Indeed, the constraint $P(A_j x \leq b_j) \geq 1 - \varepsilon_j$ leads to: $\exp\left(-\frac{2d_j(x)^2}{\sum_{i \in I_u(j)} \delta_{ji}^2 x_i^2}\right) \leq \varepsilon_j \Leftrightarrow -2d_j(x)^2 \leq \ln(\varepsilon_j).\sum_{i \in I_u(j)} \delta_{ji}^2 x_i^2$. Since all the variables are boolean, we have: $x_i^2 = x_i$ for all $i \in I$, and we can classically introduce for all $(k,l) \in I^2$ with $k < l$: $y_{kl} = x_k x_l$, adding constraints $y_{kl} \leq x_k$ and $y_{kl} \leq x_l$, plus the implicit non-negativity constraints $y_{kl} \geq 0$.

Note also that it is necessary to add some constraints defining the domain of validity of the used bound: $\mu_j(x) + \sum_{i \notin I_u(j)} A_{ji} x_i \leq b_j$ for all $j \in J$. In the case of bound (5.1), such a linearization implies the adding of $n(n-1)/2$ new non-integer variables, as well as $n(n-1) + m$ new constraints.

Hence, with respect to approximation bound (5.1), the separate probability constraints problem (2.4) can be approximated through:

$$\begin{array}{lll} \max & cx & \\ \text{s.t.} & \sum_{i \in I} \left(4b_j E[A_{ji}] - 2E[A_{ji}]^2 - \ln(\varepsilon_j)\delta_{ji}^2\right).x_i & \\ & \quad -2\sum_{(k,l) \in I^2, k \neq l} E[A_{jk}].E[A_{jl}].y_{kl} \leq 2b_j^2, & \forall j \in J \\ & y_{kl} \leq x_k, & \forall (k,l) \in I^2, k \neq l \\ & y_{kl} \leq x_l, & \forall (k,l) \in I^2, k \neq l \\ & x \in \{0,1\}^n, y \geq 0 & \end{array} \tag{5.3}$$

with the clear convention: $\delta_{ji} = 0$ when $i \notin I_u(j)$. Such a model will be referred to as an Hoeffding approximation of a problem with separate chance constraints.

Sampling methods. If the probability distributions are known, the use of sampling techniques may lead to good estimations of $P(Ax \leq b)$. From the weak law of large numbers (*cf.* for instance [20]), the sampling size can be computed so as to ensure a probabilistically good assessment of $P(Ax \leq b)$. More precisely, we have:

Lemma 5.2. *Let $x \in I\!R^n$, we denote $\Psi = P(Ax \leq b)$. Let $\mu_1, \ldots, \mu_k$ be a random sample from density $I\!I_{\{A:Ax\leq b\}}$, we denote: $\bar{\mu} = \sum_{p=1}^{k} \mu_p/k$. Furthermore, let us introduce $D = |\bar{\mu} - 1 + \varepsilon|$:*

- *if $\bar{\mu} > 1 - \varepsilon$, we have: $P\Big(\Psi < 1 - \varepsilon\Big) \leq \dfrac{1}{4kD^2}$;*
- *if $\bar{\mu} < 1 - \varepsilon$, we have: $P\Big(\Psi > 1 - \varepsilon\Big) \leq \dfrac{1}{4kD^2}$.*

Proof. Let us denote: $T = 1\!\!1_{\{A:Ax\leq b\}}$. Observe that: $\Psi = P(Ax \leq b) = E[T]$. Suppose that $\bar{\mu} > 1 - \varepsilon$, then: $P\Big(\Psi < 1 - \varepsilon\Big) \leq P\Big(\mid \Psi - \bar{\mu} \mid > D\Big)$. Then, from the weak law of large numbers: $P\big(\Psi < 1 - \varepsilon\big) \leq \sigma^2/(kD^2)$, where σ^2 is the variance of T. Furthermore, we have: $\sigma^2 = \mathbb{E}[T^2] - \mathbb{E}[T]^2 = \mathbb{E}[T^2] - \Psi^2$. Since T takes values in $\{0, 1\}$, we obtain: $\sigma^2 = \mathbb{E}[T] - \Psi^2 = \Psi(1 - \Psi) \leq 1/4$.

The case $\bar{\mu} < 1 - \varepsilon$ is exactly similar. □

Hence, if the sample size k is large enough, $\bar{\mu}$ and $P(Ax \leq b)$ are either both more than $1 - \varepsilon$, or both less than $1 - \varepsilon$, with high probability. One practical advantage of this method is that correlations between random variables can very easily be taken into account by simulating events accordingly.

Gaussian approximations. A third way to assess the probability $P(Ax \leq b)$ is to rely on normal distributions assumptions. Hence, each random variable A_{ji} is assumed to have a normal distribution, defined so that the probability $P(A_{ji} \notin [\underline{A}_{ji}, \overline{A}_{ji}])$ is small enough. This approach makes exact calculations possible (*cf.* *e.g.* [7,17]), but may often be too constraining from a model point of view. However, [9] has proposed interesting extensions from this classical Gaussian framework to a more general class of so-called radial distributions. It is proved that separate chance constraints can be equivalently written as second-order cone convex constraints. In our specific combinatorial framework, such constraints could be linearized, using the same tricks as above for the Hoeffding bound.

5.2. Difficulties and restrictions

From now on, the emphasis is put on tractability and scalability of the resolution method. Our goal is to find a good solution provided by robust problems RILP(I,γ). That is, we look for a parameter vector $\gamma \in [0, 1]^m$ corresponding to the best solution of CCILP available from this family of problems. Note that from Theorem 3.1, finding an optimal solution of CCILP would require to consider in fact problems RILP(I',γ), with varying subset $I' \subseteq I$. This is even more complex than to enumerate all the points of $\{0, 1\}^n$. To overcome this difficulty and to remain as tractable as possible, the search is limited to problems RILP(I,γ), also denoted by RILP(γ). Note however that refinements of the proposed algorithms can be obtained by investigating several subsets I', but this would impact the method tractability.

The main fact used in our algorithmic approach is the “non-increasingness” of objective value with γ. Indeed, consider two parameters γ and γ' and the respective optimal solutions x and x': if $\gamma \geq \gamma'$, then $cx \leq cx'$, since RILP(γ') is a relaxation of RILP(γ). Despite this useful property, some difficulties remain.

Missing the optimal solution. Unfortunately, the above restriction can lead to miss the optimal solution of the chance-constrained program, as illustrated in the following example.

Example 5.1. Consider the following knapsack problem:

$$\begin{array}{rc} \max & 2x_1 + 2x_2 + 3x_3 \\ \text{s.t.} & w_1x_1 + w_2x_2 + w_3x_3 \leq 2.5 \\ & x \in \{0,1\}^3 \end{array}$$

where the weights w_i are supposed independent and uniformly distributed on their respective interval of existence: $w_1 \in [0.5, 1.5]$, $w_2 \in [0.5, 1.5]$ and $w_3 \in [1.6, 2.6]$. Suppose that we want to solve the associated chance-constrained problem with $\varepsilon = 0.1$.

By enumerating all the solutions and calculating their respective feasibility probabilities, it comes that the optimal solution x^* of the chance-constrained problem is: $x^* = (0, 0, 1)$, with associated value $V^* = 3$ and feasibility probability $\mathrm{P}^* = 0.9$ (note that $\mathrm{P}^* = 1 - \varepsilon$).
Now, let us write the robust problem for $\gamma \in [0, 1]$:

$$\begin{array}{rc} \max & 2x_1 + 2x_2 + 3x_3 \\ \text{s.t.} & 0.5x_1 + 0.5x_2 + 1.6x_3 + \min\{x_1 + x_2 + x_3, 3\gamma\} \leq 2.5 \\ & x \in \{0,1\}^3. \end{array}$$

It can be checked that:

- if $0.4 < 3\gamma \leq 1.5$, the robust solution is $(1, 1, 0)$, with associated value $V = 4$ and feasibility probability $\mathrm{P} = 7/8 < 0.9$ (easy to calculate from classical probability results on the sum of two random variables).
- if $3\gamma > 1.5$, the robust solution is $(1, 0, 0)$, with associated value $V = 2$ and feasibility probability $\mathrm{P} = 1$.

Hence, the optimal solution x^* is an optimal solution for none of the robust problems tested. However, observe that x^* is an optimal solution of RILP(I',γ) when $I' = \{3\}$ and $\gamma = 0$.

Non-monotonicity of the feasibility probability. On the other hand, it would be useful to characterize the behavior of the feasibility probability $P(Ax \leq b)$. More precisely, for a given $\gamma \in [0, 1]^m$, let us denote by $x(\gamma)$ an optimal solution of RILP(γ): it could seem natural that when $\gamma \leq \gamma'$, then $P\big(Ax(\gamma) \leq b\big) \geq$

$P\big(Ax(\gamma') \leq b\big)$. Unfortunately, this is not the case, as shown in the example below. This makes the use of dichotomy-type search hardly reliable when looking for a good robust parameter γ.

Example 5.2. Consider the following knapsack problem:

$$\begin{array}{rc} \max & 2x_1 + 2x_2 + 3x_3 \\ \text{s.t.} & w_1x_1 + w_2x_2 + w_3x_3 \leq 3.5 \\ & x \in \{0,1\}^3 \end{array}$$

where the weights w_i are supposed independent and uniformly distributed on their respective interval of existence: $w_1 \in [1,2]$, $w_2 \in [1,2]$ and $w_3 \in [1.5, 5.5]$. The associated robust problem for $\gamma \in [0,1]$ is:

$$\begin{array}{rc} \max & 2x_1 + 2x_2 + 3x_3 \\ \text{s.t.} & x_1 + x_2 + 1.5x_3 + \min\{x_1 + x_2 + 4x_3, 6\gamma\} \leq 3.5 \\ & x \in \{0,1\}^3. \end{array}$$

The robust solutions can be described for all values of γ:

- if $\gamma = 0$, the robust solution is $(1,1,1)$, with value $V = 7$ and feasibility probability $P = 0$;
- if $0 < 6\gamma \leq 1$, the robust solution is $(1,0,1)$, with value $V = 5$. The feasibility probability can be calculated from the classical probability results about sums of random variables (rvs): $P = 1/8$;
- if $1 < 6\gamma \leq 1.5$, the robust solution is $(1,1,0)$, with value $V = 4$ and (after calculations) feasibility probability $P = 7/8$;
- if $1.5 < 6\gamma \leq 2$, the robust solution is $(0,0,1)$, with value $V = 3$ and feasibility probability $P = 0.5$;
- if $2 < 6\gamma$, the robust solution is $(0,0,0)$, with value $V = 0$ and feasibility probability $P = 1$.

This enumeration shows that the feasibility probability of solutions obtained from RILP(γ) is not non-decreasing with γ for the above simple uncertain knapsack problem.

5.3. Algorithms

Despite these difficulties, the robust framework remains practically attractive. We propose hereafter an algorithm to obtain good solutions to chance-constrained combinatorial problems in a tractable way. But let us first state a useful result:

Lemma 5.3. *Let x^* be an optimal solution of RILP(γ). Then, x^* is also optimal for RILP(γ'), where for each $j \in J$:*

- *if $\sum_{i \in I_u(j)} \delta_{ji} x_i^* \leq \gamma_j \sum_{i \in I_u(j)} \delta_{ji}$: $\gamma'_j = 1$;*
- *otherwise: $\gamma'_j = \big(b_j - \sum_{i \in I} \underline{A}_{ji} x_i^*\big) / \sum_{i \in I_u(j)} \delta_{ji}$.*

Proof. That x^* is feasible for RILP(γ') is immediate. Moreover, since $\gamma' \geq \gamma$, RILP(γ) is a relaxation of RILP(γ'). Thus, x^* is optimal for RILP(γ'). □

Hence, it is useless to investigate robust parameters "between" γ and γ', since they have optimal solutions in common.

Lemma 5.4. *Suppose that the* $\{\underline{A}_{ji}\}_{i\in I}$ *are integers. Let* $\gamma \in [0,1]^m$ *and* γ' *computed as explained in Lemma 5.3. Consider* γ'' *equal to* γ'*, except:* $\gamma''_j = \min\left\{1, \gamma'_j + 1/\sum_{i\in I_u(j)} \delta_{ji}\right\}$ *for one given index* $j \in J$ *such that* $\gamma'_j < 1$*. Let* x'' *be an optimal solution of RILP*(γ'')*. For any* $g \neq \gamma'$ *such that* $\gamma' \leq g \leq \gamma''$*,* x'' *is an optimal solution of RILP*(g)*.*

Proof. Let x be an optimal solution associated to γ, we define γ' from Lemma 5.3, assuming that there exists $j \in J$ such that $\gamma'_j < 1$: $\sum_{i\in I} \underline{A}_{ji} x_i + \gamma'_j \sum_{i\in I_u(j)} \delta_{ji} = b_j$. Consider any $g \in [\gamma', \gamma'']$, $g \notin \{\gamma', \gamma''\}$, and let us denote by $\hat{x}$ the associated optimal solution. Let us prove that $\hat{x}$ is feasible for RILP(γ''). If $\sum_{i\in I} \underline{A}_{ji}\hat{x}_i + \sum_{i\in I_u(j)} \delta_{ji}\hat{x}_i \leq b_j$, this is clear. If $\sum_{i\in I} \underline{A}_{ji}\hat{x}_i + g_j \sum_{i\in I_u(j)} \delta_{ji} \leq b_j$, we have: $\sum_{i\in I} \underline{A}_{ji}\hat{x}_i - \sum_{i\in I} \underline{A}_{ji} x_i \leq (\gamma'_j - g_j)\sum_{i\in I_u(j)} \delta_{ji} < 0$. As all coefficients $\{\underline{A}_{ji}\}_{i\in I}$ have been supposed integral, this means in fact: $\sum_{i\in I} \underline{A}_{ji}\hat{x}_i - \sum_{i\in I} \underline{A}_{ji} x_i \leq -1$, or equivalently: $\sum_{i\in I} \underline{A}_{ji}\hat{x}_i + 1 \leq \sum_{i\in I} \underline{A}_{ji} x_i$. Then: $\sum_{i\in I} \underline{A}_{ji}\hat{x}_i + \gamma'_j \sum_{i\in I_u(j)} \delta_{ji} + 1 \leq \sum_{i\in I} \underline{A}_{ji} x_i + \gamma'_j \sum_{i\in I_u(j)} \delta_{ji} = b_j$. As a direct consequence, $\hat{x}$ is a feasible solution of RILP(γ'').

Then, it has been proved that $\hat{x}$ is a feasible solution of RILP(γ''). Since $\gamma'' \geq g$, $\hat{x}$ is in fact optimal for RILP(γ''). Hence, RILP(γ'') and RILP(g) have the same optimal value. Furthermore, x'' is feasible for RILP(g), since $g \leq \gamma''$. Thus, x'' is optimal for RILP(g). □

This shows that it is also useless to consider the robust parameters "between" γ' and γ''. Lemma 5.3 and 5.4 are used in the following algorithm:

```
Fast algorithm
Step 0:  Set $\gamma = 0$.
Step 1:  Solve RILP$(\gamma)$, and let $x$ be the corresponding optimal solution.
Step 2:  Compute (or bound) $P(Ax \leq b)$.
Step 3:  If $P(Ax \leq b) < 1 - \varepsilon$:
             compute $\gamma' \geq \gamma$ as described in Lemma 5.3, and set $\gamma \leftarrow \gamma'$;
             select $j$ such that: $j = \arg\min_{k\in J:\gamma'_k<1} \left\{b_k - \sum_{i\in I} E[A_{ki}]x_k\right\}$;
             set $\gamma_j \leftarrow \gamma_j + 1/\sum_{i\in I_u(j)} \delta_{ji}$ and go to Step 1.
Step 4:  If $P(Ax \leq b) \geq 1 - \varepsilon$: STOP.
```

The idea is to investigate an increasing sequence of parameters γ. At each loop, the current robust problem is solved (Step 1). Then, the feasibility probability of the obtained solution is computed (Step 3). If this solution is feasible for CCILP, the algorithm stops (Step 4). Otherwise, γ is increased according to Step 3. Note that this latter process ensures that the algorithm stops in finite time with a feasible solution for CCILP. The number of iterations is upperly bounded by $\sum_{j\in J} \left\lceil \sum_{i\in I_u(j)} \delta_{ji} \right\rceil$; in practice, it is much lower.

Some more explanations are needed for Step 3. The index j is chosen with the aim to improve effectively the feasibility of a solution. Hence, in the proposed implementation, we increase the protection term corresponding to the smallest gap between right hand side and the expected value of left hand side. This is motivated by observing that many probability inequalities involve this gap (*cf.* the Markov inequality, the Hoeffding inequality...). But other more relevant criteria may be thought of, depending on the specific problem dealt with. Suppose for instance that all data A and b are positive, inspired by the Markov inequality, we could take: $j = \arg\min_{k\in J}\left\{1 - (\sum_{i\in I} E[A_{ki}]x_k)/b_k\right\} = \arg\max_{k\in J}\left\{(\sum_{i\in I} E[A_{ki}]x_k)/b_k\right\}$. In our practical tests, we observed that the criterion chosen did not have a significant impact on results.

The fast algorithm described above may still be difficult in practice, due to the exact resolution of RILP, for instance through the integer linear program (4.2). A natural idea is to replace the protection term $\Delta_j(x)$ by the constant: $\Delta'_j(x) = \gamma_j \cdot \sum_{i\in I_u(j)} \delta_{ji}$. Observe that this latter option corresponds to change simply the right hand side b_j. When this constant protection term becomes “too large”, the constraint j may be replaced by the worst case: $\overline{A}_j x \le b_j$. These ideas are precisely described in the following algorithm:

Alternative fast algorithm

```
Step 0:  Set $\gamma = 0$.
Step 1:  Let $x$ be an optimal solution of the problem:
           max   $cx$
           s.t.  $\sum_{i\in I_u(j)} \underline{A}_{ji} x_i + \gamma_j \cdot \sum_{i\in I_u(j)} \delta_{ji} + \sum_{i\notin I_u(j)} A_{ji} x_i \le b_j, \ \forall j \in J$
                 $x \in \{0,1\}^n$
Step 2:  Compute (or bound) $P(Ax \le b)$.
Step 3:  If $P(Ax \le b) < 1 - \varepsilon$:
             compute $\gamma' \ge \gamma$ as described in Lemma 5.3, and set $\gamma \leftarrow \gamma'$;
             for each $j \in J$, if $\gamma'_j = 1$, then replace the corresponding
             constraint with:
                 $\sum_{i\in I_u(j)} \overline{A}_{ji} x_i + \sum_{i\notin I_u(j)} A_{ji} x_i \le b_j$;
             select $j$ such that: $j = \arg\min_{k\in J:\gamma'_k<1}\left\{b_k - \sum_{i\in I} E[A_{ki}]x_i\right\}$;
             set $\gamma_j \leftarrow \gamma_j + 1/\sum_{i\in I_u(j)} \delta_{ji}$ and go to Step 1.
Step 4:  If $P(Ax \le b) \ge 1 - \varepsilon$: STOP.
```

In this alternative algorithm, at each loop, only one nominal ILP problem has to be solved (*cf.* Step 1). The simplicity and tractability of the proposed algorithms appear clearly. Furthermore, by contrast with many meta-heuristic based methods, no specific tuning is necessary here. This is especially important in practice, since a lot of time is usually spent looking for good parameters of meta-heuristic.

5.4. Computing upper bounds for CCILP

So far, heuristics have been designed to obtain feasible solutions for CCILP. In order to be able to assess the quality of the provided solutions, an upper bound on the optimal value of CCILP is needed. An obvious bound is obtained by considering that all the uncertain coefficients take their minimal value, that corresponds to the best scenario. This can be improved with the following result:

Proposition 5.1. *For all $j \in J$, let us introduce:*

- $p_j \geq 0$ *such that, for any optimal solution* x^* *of CCILP:* $p_j \leq \sum_{i \in I_u(j)} x_i^*$*;*
- $I'_j \subseteq I_u(j)$ *such that* $|I'_j| = p_j$ *and:* $\max\{\delta_{ji} \mid i \in I'_j\} \leq \min\{\delta_{ji} \mid i \in I_u(j) \backslash I'_j\}$.

Suppose that the $\{\eta_{ji}\}_{i \in I_u(j)}$ *are i.i.d. Then, consider* $\gamma \in [0,1]^m$ *such that, for all* $j \in J$*:*

$$P\left(\sum_{i \in I'_j} \delta_{ji}\eta_{ji} \leq \gamma_j \sum_{i \in I_u(j)} \delta_{ji}\right) < 1 - \varepsilon.$$

Any optimal solution x^* *of CCILP satisfies, for all* $j \in J$*:*

$$\sum_{i \in I} \underline{A}_{ji} x_i^* + \gamma_j \sum_{i \in I_u(j)} \delta_{ji} \leq b_j.$$

Proof. Let x^* be an optimal solution of CCILP. We prove by contradiction that x^* satisfies the above inequalities. Suppose that there exists $j \in J$ such that: $\sum_{i \in I} \underline{A}_{ji} x_i^* + \gamma_j \sum_{i \in I_u(j)} \delta_{ji} > b_j$. Then:

$$\begin{aligned} P(A_j x^* \leq b_j) &\leq P\left(A_j x^* \leq \underline{A}_j x^* + \gamma_j \textstyle\sum_{i \in I_u(j)} \delta_{ji}\right) \\ &= P\left(\textstyle\sum_{i \in I_u(j)} \delta_{ji}\eta_{ji} x_i^* \leq \gamma_j \sum_{i \in I_u(j)} \delta_{ji}\right) \\ &\leq P\left(\textstyle\sum_{i \in I'_j} \delta_{ji}\eta_{ji} \leq \gamma_j \sum_{i \in I_u(j)} \delta_{ji}\right). \end{aligned}$$

The latter inequality comes from the fact that the $\{\eta_{ji}\}_{i \in I_u(j)}$ are i.i.d., and from the definition of I'_j. Then, observe that: $P(Ax^* \leq b) \leq P(A_j x^* \leq b_j) < 1 - \varepsilon$. This is a contradiction. □

Often, the $\{p_j\}_{j \in J}$ can be computed quite easily. From p_j, the corresponding set I'_j is built in linear time by considering the smallest variations δ_{ji} for $i \in I_u(j)$. Then, the γ_j may be found by dichotomy. Note that to reach upper bounds as good as possible, we have to find γ as large as possible.

When dealing with separate chance constraints (2.4), Proposition 5.1 can be easily adapted. It is sufficient to consider γ such that, for all $j \in J$:

$$P\left(\textstyle\sum_{i \in I'_j} \delta_{ji}\eta_{ji} \leq \gamma_j \sum_{i \in I_u(j)} \delta_{ji}\right) < 1 - \varepsilon_j.$$

In this latter case, and when only one coefficient per row is subject to uncertainty, the bound obtained from Proposition 5.1 is the optimal value of CCILP if γ is chosen large enough. (This occurs for instance when right hand side data only are uncertain.)

Proposition 5.1 also enables us to accelerate again the Fast Algorithm. Indeed, instead of beginning with $\gamma = 0$ at Step 0, we can begin directly with a parameter

γ satisfying the requirements of Proposition 5.1. This pre-optimization procedure avoids useless iterations.

Finally, when relying on the Hoeffding bound to assess the feasibility probability, the analytical expression of the chance constrained problem may provide upper bounds, by relaxing integrality constraints or considering some lagrangean relaxations, for instance.

Other upper bounds have been investigated, see the long version of the current paper [19]. Despite their theoretical interest, they did not lead to significant improvements.

6. Applications and numerical tests

The framework proposed is suitable for a wide range of application problems, such as:

- the 0–1 knapsack problem with uncertain weights, as well as its classical extensions, multiple knapsack and multi-dimensional knapsack problems;
- shortest paths problems with uncertain edge weights;
- unsplittable multicommodity flow problems with uncertain demands;
- some capacitated location problems.

Hereafter, numerical results will be provided for the knapsack and multi-dimensional knapsack problems with uncertain weights. The computer programs have been written in C++ with CPLEX 9.0. All the tests have been performed on a computer running an Intel(R) Xeon(TM) processor 2.8 GHz with 2 Go of RAM.

6.1. Instances and resolution methods

Two different types of instances have been generated. With the first ones, all weight variations δ_{ji} are 10% of the minimal weight ("proportional variations"). With the second type of instances, these weight variations are randomly chosen in an interval. This means that the variations are not correlated to the minimal coefficient value ("uncorrelated variations").

For each type of problem, two different series of tests have been performed, the first one using sampling methods to estimate the feasibility probability, and the second one relying on the Hoeffding probability bound (see Sect. 5.1). When dealing with multi-dimensional knapsack problems, tests have been run for separate and joint probability constraints (*cf.* Sect. 2.3).

When considering separate probability constraints, several resolution methods are compared on each problem.

- **Worst**: it corresponds to the worst case value, that is the value when all coefficients A_{ji} take their maximal value $\overline{A}_{ji}$.
- **M1**: the first one, denoted by M1, relies on the ideas of [5], where the robust model RILP$_2$ (*cf.* Eqs. (4.3)) has been introduced. The authors pointed out the fact that, given a parameter vector Γ, the feasibility probability of any optimal solution x of RILP$_2(\Gamma)$ could be bounded by: $\forall j \in J, P\left(A_j x \leq b_j\right) \geq P\left(\sum_{i \in I_u(j)} \eta_{ji} \leq \Gamma_j\right)$. Hence, for each $j \in J$, we

look for the lowest Γ_j satisfying: $P\left(\sum_{i\in I_u(j)}\eta_{ji} \leq \Gamma_j\right) \geq 1-\varepsilon_j$. Then, the problem $\text{RILP}_2(\Gamma)$ is solved. Observe that only one robust problem needs to be solved with this approach.

- **M2**: a second resolution method, denoted by M2, has been directly inspired by the work of [18], that also relies on the robust model RILP_2. Starting with $\Gamma = 0$, a sequence of increasing parameter vectors Γ is progressively generated. At each new Γ, the corresponding robust problem $\text{RILP}_2(\Gamma)$ is solved. The choice of a new parameter vector exploits ideas very similar to those exposed in Section 5.3 for the fast algorithm: from the current Γ and the associated robust solution, a new vector $\Gamma' \geq \Gamma$ is built, which corresponds to the same optimal value. Then, its lowest component Γ_j is increased by $1/\max_{i\in I_u(j)}\delta_{ji}$. For more technical details, we refer to [18], where the same ideas are introduced on the restricted case of the knapsack problem.
- **FA**: a third approach is the fast algorithm, as introduced in Section 5.3.
- **AFA**: this fourth method is the alternative fast algorithm of Section 5.3.
- **Optimum**: when possible, the optimal solutions have also been computed. When using the Hoeffding bound with separate probability constraints, the optimal solution can be found through the models developed in Section 5.1; this approach will be denoted by HA. In the other cases, with sampling estimation or joint probability constraints, all solutions are enumerated. Of course, this is possible only for small size instances.
- **Upper bound**: when using the sampling estimation of the feasibility probability, an upper bound on the optimal value of CCILP is computed according to Proposition 5.1. Since we deal with knapsack problems, each value p_j is computed in linear time by filling knapsack j with the largest maximal coefficients $\overline{A}_{ji}$ first.
- **Best**: it corresponds to the best case value, that is the value when all coefficients A_{ji} take their minimal value $\underline{A}_{ji}$.

The approaches "worst", "M1", "M2", "FA", "AFA" and "optimum" provide feasible solutions to the chance-constrained problem. On the contrary, "upper bound" and "best" correspond only to bounds on the optimal value investigated; that is the reason for writing the corresponding results in italic in the result tables.

When dealing with joint probability constraints, method M1 is no more usable. Furthermore, as the Hoeffding approximation model cannot be linearized with joint probability constraints, it has not been implemented.

The solution values obtained are always compared to the worst case value.

6.2. The knapsack problem

This problem is one of the simplest applications, since it contains only one constraint:

$$\begin{array}{rl} \max & \sum_{i\in I} p_i x_i \\ \text{s.t.} & \sum_{i\in I} w_i x_i \leq c \\ & x \in \{0,1\}^n \end{array}$$

$c > 0$ denotes the knapsack capacity. For each i, $p_i > 0$ is the profit of taking element i, while $w_i > 0$ is the weight of this element. In the following, all the weights are assumed to be unprecisely known: $\forall i \in I, w_i \in [\underline{w}_i, \underline{w}_i + \delta_i]$.

Following the ideas of Section 5.3, the chance-constrained problem will be heuristically solved through a sequence of robust problems. With clear notations, RKP(γ) will denote the robust version of the uncertain knapsack problem for $\gamma \in [0,1]$, while CCKP will be its chance-constrained version. According to Theorem 4.1, solving RKP(γ) requires to solve only 2 knapsack problems, namely:

$$\begin{aligned} & (a)\ \max \left\{ p \cdot x | \overline{w} \cdot x \leq c, x \in \{0,1\}^n \right\} \\ \text{and:}\quad & (b)\ \max \left\{ p \cdot x | \underline{w} \cdot x + \gamma \textstyle\sum_{i \in I} \delta_i \leq c, x \in \{0,1\}^n \right\}. \end{aligned}$$

The optimal solution of RKP(γ) is obtained as the best solution among this of (a) and this of (b). The problem (a) can be solved only once at the beginning, the problem (b) being solved for each value of γ tested. Hence, the fast algorithm proposed in Section 5.3 can be very easily used here:

```
Chance-constrained knapsack algorithm (CCKA)
Step 0: Solve problem $(a)$, and let $\overline{x}$ be its optimal solution. Set $\gamma = 0$.
Step 1: Solve $(b)$, and let $x$ be the corresponding optimal solution.
Step 2: If $p.x \leq p.\overline{x}$: set $x = \overline{x}$ and STOP.
Step 3: Estimate or bound $P(Ax \leq b)$.
Step 4: If $P(Ax \leq b) < 1 - \varepsilon$:
            compute $\gamma' \geq \gamma$ as described in Lemma 5.3;
            set $\gamma = \gamma' + 1/\sum_{i \in I} \delta_i$ and go to Step 1.
Step 5: If $P(Ax \leq b) \geq 1 - \varepsilon$: STOP.
```

Some of the results from [18] can be adapted here, showing that algorithm CCKA may provide optimal solutions to chance-constrained problems in some cases. Let $x \in \{0,1\}^n$, we say that x is *non-increasing* if: $i < j \Rightarrow x_i \geq x_j$. This means that x looks like: $1, 1, \ldots, 1, 0, \ldots, 0$.

Proposition 6.1. *Suppose that weights and profits can be sorted so that:*

$$i < j \Rightarrow \begin{cases} p_i > p_j \\ \underline{w}_i \leq \underline{w}_j \\ \delta_i \leq \delta_j. \end{cases} \tag{6.1}$$

Moreover, suppose that random variables (rvs) $\{\eta_i\}_{i \in I}$ are independent and identically distributed (i.i.d.). Then, there exists $\gamma^* \in [0,1]$ such that an optimal solution of RKP(γ^*) is also optimal for CCKP.

This result is close to Theorem 2 in [18]. A specific proof can be found in the long version [19].

Corollary 6.1. *Suppose that conditions (6.1) hold and that the $\{\eta_i\}_{i \in I}$ are i.i.d. Let $\gamma \in [0,1]$, the optimal solution x^* of RKP(γ) is an optimal solution of CCKP with $\varepsilon = P(wx^* \geq c)$.*

Proof. Let $\varepsilon = P(wx^* \geq c)$. We know from Proposition 6.1 that there exists γ' such that an optimal solution x' of RKP(γ') is optimal for CCKP. By construction, x^* is feasible for CCKP. Since both vectors x' and x^* are non-increasing, we have: $x' \geq x^*$ (otherwise, $px' < px^*$). If $x' \neq x^*$, we have then: $P(wx' \leq c) < P(wx^* \leq c) = \varepsilon$. This means that $x' = x^*$, that shows that x^* is optimal for CCKP. □

Thus, under condition (6.1), the algorithm CCKA can be used to obtain an optimal solution of CCKP. It is sufficient, for each value γ tested, to consider non-increasing solutions. This is not of great practical interest, since under conditions (6.1), the problem can be easily solved by adding successively the elements of largest profit, until the probability condition is reached. However, the above results indicate the relevance of algorithm CCKA, at least when dealing with some particular types of instances.

To build the test instances, the minimal weights $\{\underline{w}_i\}_{i\in I}$, as well as the profits $\{p_i\}_{i\in I}$, are randomly chosen in [100, 1000], and the capacity c is randomly chosen between $\sum_{i\in I} \underline{w}_i/3$ and $2\sum_{i\in I} \underline{w}_i/3$. All data c, $\underline{w}_i$ and δ_i are positive integers.

When using sampling methods to estimate the feasibility probability of a given solution, the rvs $\{w_i\}_{i\in I}$ are supposed independent and uniformly distributed on their definition intervals. We rely on Lemma 5.2 to ensure that we have a feasible solution for CCKA with probability at least 99.99%. The average probabilities appearing in result tables are then approximative. When using the Hoeffding probability bound (5.1), the rvs $\{w_i\}_{i\in I}$ are supposed independent and centered: $\forall i \in I, E[w_i] = \underline{w}_i + \delta_i/2$. The goal is to find a good knapsack filling being feasible with probability at least 90%, which means that $\varepsilon = 0.1$.

In the "proportional variations" case, we set: $\forall i \in I, \delta_i = 0.1\underline{w}_i$ (δ_i is possibly rounded to be integral). With "uncorrelated variations", the $\{\delta_i\}_{i\in I}$ are randomly chosen in [10, 100].

Results when using the sampling method. Instances with $n = 25$, $n = 100$ and $n = 200$ have been considered; for each size, one hundred instances have been run. The optimal solution, found through a complete enumeration of all possible solutions, is given only for $n = 25$. Some major trends can be underlined from Table 1, which reports the average results for each case. First, M1 leads to very small improvements: while being very fast, this algorithm remains far less interesting than M2 and CCKA. Secondly, M2 and CCKA provide very close results. CCKA appears as much faster than M2, that is due to the very easy resolution of RKP(γ) through one knapsack problem. Thirdly, M2 and CCKA provide both near-optimal solutions for $n = 25$. More precisely, we observed that M2 always found the optimum for the case of proportional variations (see Tab. 3).

The careful observation of the obtained results leads to conclude that CCKA provides slightly better solutions than M2 for proportional variations, while the contrary occurs for uncorrelated variations (see also Tab. 2, which gives the number of instances where M2 or CCKA provided the best solution). This may be explained from the protection terms used by robust models RILP and RILP$_2$. For RILP, $\Delta(x) = \min\left\{\gamma \cdot \sum_{i\in I} \delta_i, \sum_{i\in I} \delta_i x_i\right\}$; for RILP$_2$, $\Delta(x)$ is replaced by

Table 1. CCKP: Average results on 100 instances, when using the sampling method to assess the feasibility probability.

			Average value	Average improvement %	Average proba.	Average time (s)
Proportional variations	$n = 25$	worst	9567.89	–	1	0.01
		M1	9578.7	0.11	1.00	0.02
		M2	9748.77	1.89	0.97	0.05
		CCKA	9750.34	1.91	0.97	0.03
		optimum	9751.09	1.91	0.97	12.99
		upper bound	9987.68	4.39	–	0.00
		best	10062.8	5.17	–	0.00
	$n = 100$	worst	38682.04	–	1	0.06
		M1	38735.06	0.14	1.00	0.14
		M2	39566.08	2.29	0.93	3.54
		CCKA	39568.24	2.29	0.93	0.91
		upper bound	40445.15	4.56	–	0.01
		best	40738.43	5.32	–	0.01
	$n = 200$	worst	77750.28	–	1	0.15
		M1	77868.61	0.15	1.00	0.38
		M2	79582.06	2.36	0.92	30.00
		CCKA	79586.44	2.36	0.92	5.41
		upper bound	81280.36	4.54		0.01
		best	81859.18	5.28		0.01
Uncorrelated variations	$n = 25$	worst	9467.69	–	1	0.01
		M1	9473.99	0.07	1.00	0.02
		M2	9709.92	2.56	0.97	0.06
		CCKA	9709.23	2.55	0.97	0.04
		optimum	9709.62	2.56	0.97	12.94
		upper bound	10012.3	5.75	–	0.00
		best	10094.73	6.62	–	0.00
	$n = 100$	worst	38657.14	–	1	0.05
		M1	38714.65	0.15	1.00	0.14
		M2	39749.17	2.82	0.93	4.19
		CCKA	39744.02	2.81	0.93	1.12
		upper bound	40879.29	5.75	–	0.01
		best	41190.37	6.55	–	0.01
	$n = 200$	worst	77432.07	–	1	0.18
		M1	77545.8	0.15	1.00	0.5
		M2	79708.09	2.94	0.92	40.52
		CCKA	79700.84	2.93	0.92	7.64
		upper bound	81898.31	5.77	–	0.02
		best	82503.48	6.55	–	0.02

$\Delta_2(x) = \max\left\{\sum_{i\in S}\delta_i x_i : S \subseteq I, |S| = \Gamma\right\}$, when Γ is integral. It can be seen that Δ_2 will always penalize solutions involving coefficients with large variations. Hence, for instance, if we have two elements i and j such that $p_i = p_j$, $\underline{w}_i = \underline{w}_j$ and $\delta_i < \delta_j$, RILP$_2$ will "prefer" element i to element j, although RILP will make

Table 2. CCKP: Number of instances where CCKA performs better or worse than M2, when using the sampling method to assess the feasibility probability.

		CCKA > M2	CCKA = M2	CCKA < M2
Proportional variations	$n = 25$	2%	98%	0%
	$n = 100$	20%	71%	9%
	$n = 200$	38%	50%	12%
Uncorrelated variations	$n = 25$	0%	98%	2%
	$n = 100$	11%	69%	20%
	$n = 200$	9%	37%	54%

Table 3. CCKP: Number of instances where CCKA or M2 provide the optimal solution, when using the sampling method.

	CCKA = optimum	M2 = optimum
Proportional variations, $n = 25$	100%	98%
Uncorrelated variations, $n = 25$	96%	98%

no difference between them most of the time. Indeed, with Δ, the variations are taken into account mainly through their sum $\sum_{i\in I}\delta_i$, since most of the time: $\Delta(x) = \gamma \cdot \sum_{i\in I}\delta_i$. This may explain the better ability of RILP$_2$ to select good solutions when variations are not correlated to minimal weights.

Finally, upper bounds computed from Proposition 5.1 improve on the best case value. However, they remain very weak. It appears that obtaining good upper bounds on the optimal value of CCILP may require more sophisticated algorithms.

Results when using the Hoeffding bound. As before, one hundred instances have been tested for each size $n = 50$, $n = 100$ and $n = 200$. From Table 4, we see that the main observations detailed here-above remain true: M1 leads to poor solutions; M2 and CCKA lead to similar quality solutions, but CCKA is faster than M2. When $n = 50$, a comparison with the optimal solution, given by HA, can be done. It appears that M2 and CCKA lead to near-optimal solutions. For $n = 100$, HA has been solved by Cplex with a time limit of 30 min; as the wide majority of the instances have not been solved to optimality within this time limit, the average gap is reported in the last column of Table 4. It appears that the solutions found by this latter process are worse than those provided by M2 or CCKA in only a few seconds. As HA was so hard to solve already for $n = 100$, it has not been tested when $n = 200$.

Hence, the solutions given by CCKA and M2 are shown to be very good quality ones. More detailed comments may be done with respect to each method performance. When dealing with proportional variations, the solutions of CCKA are slightly better than those of M2 (*cf.* also Tab. 5). The contrary occurs for uncorrelated variations. The reasons for this are the same than those given when using the sampling method.

Table 4. CCKP: Average results on 100 instances, when using the Hoeffding bound to assess the feasibility probability.

			Average value	Average improvement %	Average proba.	Average time (s) or gap (%)
Proportional variations	$n = 50$	Worst	19990.13	–	1	0.01 s
		M1	19991.64	0.01	1.00	0.02 s
		M2	20268.33	1.39	0.94	0.41 s
		CCKA	20268.32	1.39	0.94	0.12 s
		HA	20268.54	1.39	0.94	50.03 s
	$n = 100$	Worst	38939.2	–	1	0.05 s
		M1	38954.94	0.04	1.00	0.05 s
		M2	39614.8	1.74	0.92	4.15 s
		CCKA	39617.11	1.74	0.92	0.79 s
		HA	39611.01	1.73	0.92	0.66%
	$n = 200$	Worst	79704.63	–	1	0.16 s
		M1	79768.07	0.08	1.00	0.21 s
		M2	81262.11	1.95	0.91	37.66 s
		CCKA	81264.83	1.96	0.91	4.92 s
Uncorrelated variations	$n = 50$	Worst	19428.92	–	1	0.02 s
		M1	19432.77	0.02	1.00	0.02 s
		M2	19766.52	1.74	0.94	0.57 s
		CCKA	19764.66	1.73	0.94	0.18 s
		HA	19767.16	1.74	0.94	218.25 s
	$n = 100$	Worst	38537.29	–	1	0.07 s
		M1	38552.72	0.04	1.00	0.07 s
		M2	39357.63	2.13	0.92	5.63 s
		CCKA	39352.04	2.11	0.92	1.13 s
		HA	39351.06	2.11	0.92	0.94%
	$n = 200$	Worst	77473.05	–	1	0.16 s
		M1	77518.12	0.06	1.00	0.23 s
		M2	79394.54	2.48	0.91	40.15 s
		CCKA	79378.69	2.46	0.91	5.97 s

Table 5. CCKP: Number of instances where CCKA performs better or worse than M2, when using the Hoeffding bound to assess the feasibility probability.

		CCKA > M2	CCKA = M2	CCKA < M2
Proportional variations	$n = 50$	1%	98%	1%
	$n = 100$	14%	84%	2%
	$n = 200$	26%	72%	2%
Uncorrelated variations	$n = 50$	0%	90%	10%
	$n = 100$	2%	76%	22%
	$n = 200$	6%	37%	57%

Table 6. CCKP: Number of instances where CCKA or M2 provide the optimal solution when using the Hoeffding bound.

	CCKA = optimum	M2 = optimum
Proportional variations, $n = 50$	99%	99%
Uncorrelated variations, $n = 50$	86%	95%

Remark 6.1. Note that the feasibility probability indicated in tables for solutions of M1 is always "1.00". In fact, this feasibility probability was not exactly 1, but too close to 1 with respect to the numerical precision of the given results.

6.3. The multi-dimensional knapsack problem

We also study the following generalization of the knapsack problem, known as the multi-dimensional Knapsack problem (MKP):

$$\begin{array}{rc} \max & \sum_{i\in I} p_i x_i \\ \text{s.t.} & \sum_{i\in I} w_{ji} x_i \le c_j, \quad \forall j \in J \\ & x \in \{0,1\}^n. \end{array}$$

We present results only for five-dimensional knapsack problems, *i.e.* $m = 5$. All data c, $\underline{w}_{ji}$ and δ_{ji} are supposed positive and integral. As before, all the weights are assumed to be uncertain, with either proportional or uncorrelated variations. All data are randomly generated according to the same rules than for knapsack problems. On the other hand, we keep similar probabilistic assumptions: when using sampling methods, the rvs $\{w_{ji}\}_{j\in J, i\in I}$ are supposed independent and uniformly distributed on their definition intervals; when using the Hoeffding bound, the rvs are supposed independent and centered.

Two chance-constrained problems are considered. The first one, denoted by CCMKP$_1$, addresses the case of separate probability constraints:

$$\begin{array}{rc} \max & \sum_{i\in I} p_i x_i \\ \text{s.t.} & P\left(\sum_{i\in I} w_{ji} x_i \le c_j\right) \ge 1 - \varepsilon_j, \quad \forall j \in J. \\ & x \in \{0,1\}^n. \end{array}$$

The second one, denoted by CCMKP$_2$, is with a joint probability constraint:

$$\begin{array}{rc} \max & \sum_{i\in I} p_i x_i \\ \text{s.t.} & P\left(\forall j \in J : \sum_{i\in I} w_{ji} x_i \le c_j\right) \ge 1 - \varepsilon. \\ & x \in \{0,1\}^n. \end{array}$$

We aim at solving these problem for $\varepsilon_j = 0.1$, for all $j \in J$, or $\varepsilon = 0.1$ in CCMKP$_2$.

When dealing with CCMKP$_1$, as with the knapsack problem, the corresponding Hoeffding approximation (HA) problem can be formulated as an integer linear program (see Sect. 5.1). Hence, the results obtained with the proposed algorithm can

Table 7. CCMKP$_1$: average results on 100 instances, when using the sampling method to assess the feasibility probability.

			Average value	Average improvement %	Average time (s)
Proportional variations	$n = 20$	Worst	6290.41	–	0.03
		M1	6290.41	0.00	0.05
		M2	6397.65	1.70	0.12
		FA	6399.13	1.73	0.12
		AFA	6399.16	1.73	0.06
		Optimal	6439.69	2.37	27.89
		Upper bound	6665.18	5.96	0.01
		Best	6720.95	6.84	0.01
	$n = 100$	Worst	32825.45	–	1.06
		M1	32825.83	0.00	1.34
		M2	33471.83	1.97	35.59
		FA	33476.27	1.98	33.9
		AFA	33478.55	1.99	6.19
		Upper bound	34707.99	5.74	0.08
		Best	34976.34	6.55	0.08
Uncorrelated variations	$n = 20$	Worst	6096.38	–	0.03
		M1	6096.38	0.00	0.04
		M2	6230.03	2.19	0.14
		FA	6232.25	2.23	0.13
		AFA	6229.99	2.19	0.07
		Optimal	6272.68	2.89	25.65
		Upper bound	6500.00	6.62	0.01
		Best	6587.48	8.06	0.01
	$n = 100$	Worst	33261.65	–	0.82
		M1	33262.35	0.00	0.98
		M2	34071.59	2.44	38.16
		FA	34058.86	2.40	40.45
		AFA	34056.91	2.39	7.67
		Upper bound	35484.68	6.68	0.06
		Best	35765.99	7.53	0.07

be compared with the best solution obtainable from our probabilistic hypothesis. For CCMKP$_2$, recall that the Hoeffding approximation would require to solve an integer non-linear problem, that is difficult and beyond the scope of this paper. On the other hand, recall that for CCMKP$_2$ (joint probability constraints), M1 cannot be used.

When using sampling methods, tests have been run for $n = 20$ and $n = 100$, with one hundred instances for each size. When considering the Hoeffding bound, one hundred instances have been generated for both $n = 50$ and $n = 100$. Average results are presented in Tables 7–9 (the results for CCMKP$_2$ with the Hoeffding

TABLE 8. CCMKP$_1$: average results on 100 instances, when using the Hoeffding bound.

			Average value	Average improvement %	Average gap	Average time (s)
Proportional variations	$n = 50$	Worst	16451.21	–	–	0.15
		M1	16451.21	0.00	–	0.16
		M2	16640.05	1.15	–	2.33
		FA	16637.62	1.13	–	2.21
		AFA	16638.28	1.14	–	0.62
		HA	16717.26	1.62	0.00%	117.3
	$n = 100$	Worst	16336.75	–	–	0.28
		M1	16336.75	0.00	–	0.30
		M2	16563.59	1.39	–	16.78
		FA	16568.38	1.42	–	16.83
		AFA	16566.81	1.41	–	2.79
		HA	16649.65	1.92	0.56%	905.37
Uncorrelated variations	$n = 50$	Worst	16160.2	–	–	0.14
		M1	16160.2	0.00	–	0.15
		M2	16362.68	1.25	–	2.80
		FA	16356.13	1.21	–	2.89
		AFA	16356.13	1.21	–	0.83
		HA	16454.76	1.82	0.02%	291.07
	$n = 100$	Worst	16456.5	–	–	0.37
		M1	16456.5	0.00	–	0.37
		M2	16739.52	1.72	–	20.84
		FA	16739.39	1.72	–	20.74
		AFA	16739.7	1.72	–	3.48
		HA	16830.41	2.27	0.69%	903.55

bound are not reported here, since they are very similar to the other ones; see [19]). As for the knapsack problem, M1 brings almost no improvement, while M2 and FA provide very similar quality results. AFA appears as a very effective algorithm, leading to solutions close to those of FA in far less time. However, it appears that both feasible solutions and upper bounds are quite far from the optimal values, although improving significantly both respectively on the worst case and the best case values. Note that when using HA, a time limit of 30 min has been imposed. Then, when instances are not solved to optimality, the gap to optimal value is indicated.

7. CONCLUSION

A new and tractable robust model has been proposed. It has been shown to be related to chance-constrained integer linear problems. Then, heuristic algorithms based on this robust model have been developed to obtain good feasible solutions to chance-constrained combinatorial problems. To the best of our knowledge, that is the first time such algorithms are developed to deal with general chance-constrained combinatorial problems. Extensive numerical tests have been performed on knapsack and multi-dimensional knapsack instances. They show

TABLE 9. CCMKP$_2$: average results on 100 instances, when using the sampling method.

			Average value	Average improvement %	Average proba.	Average time (s)
Proportional variations	$n = 20$	Worst	5986.51	–	1	0.03
		M2	6094.52	1.80	0.98	0.12
		FA	6095.89	1.83	0.98	0.12
		AFA	6094.74	1.81	0.98	0.07
		Optimal	6135.88	2.50	0.98	8.01
		Upper bound	6355.64	6.17	–	0.01
		Best	6424.05	7.31	–	0.01
	$n = 100$	Worst	32623.6	–	1	1.12
		M2	33266.81	1.97	0.95	44.29
		FA	33285.01	2.03	0.95	41.19
		AFA	33289.73	2.04	0.95	7.90
		Upper bound	34507.54	5.77	–	0.08
		Best	34788.63	6.64	–	0.09
Uncorrelated variations	$n = 20$	Worst	6067.63	–	1	0.03
		M2	6217.01	2.46	0.98	0.13
		FA	6207.69	2.31	0.98	0.13
		AFA	6209.54	2.34	0.98	0.07
		Optimal	6253.67	3.07	0.97	7.11
		Upper bound	6506.03	7.23	–	0.01
		Best	6589.36	8.60	–	0.01
	$n = 100$	Worst	32569.07	–	1	0.85
		M2	33370.85	2.46	0.94	43.93
		FA	33387.33	2.51	0.94	47.02
		AFA	33387.08	2.51	0.94	9.07
		Upper bound	34820.77	6.91	–	0.07
		Best	35105.49	7.79	–	0.07

the practical effectiveness of the algorithms. In particular, their high tractability make them usable for real-life large size problems. However, these numerical results also show that improvements are possible. Hence, future work will be devoted to obtaining better feasible solutions for chance-constrained integer linear problems, as well as improving the upper bounds proposed to provide some theoretical guarantee on the solutions.

REFERENCES

[1] R. Aringheri, *A Tabu search algorithm for solving chance-constrained programs*, Note del Polo 73, DTI – University of Milano (2005), available at http://www.crema.unimi.it/Biblioteca/SchedaNota.asp?Nota=92.

[2] A. Ben-Tal and A. Nemirovski, Robust solutions of linear programming problems contamined with uncertain data. *Math. Program. (Ser. A)* **88** (2000) 411–424.

[3] P. Beraldi and A. Ruszczyński, A branch and bound method for stochastic integer problems under probabilistic constraints. *Optim. Methods Softw.* **17** (2002) 359–382.

[4] D. Bertsimas and M. Sim, Robust discrete optimization and network flows. *Math. Program. (Ser. B)* **98** (2003) 49–71.

[5] D. Bertsimas and M. Sim, The Price of Robustness. *Oper. Res.* **52-1** (2004) 35–53.

[6] L. Bianchi, M. Dorigo, L.M. Gambardella and W.J. Gutjahr, *Metaheuristics in stochastic combinatorial optimization: a survey*, technical report IDSIA-08-06, IDSIA, available at www.idsia.ch/idsiareport/IDSIA-08-06.pdf (2006).

[7] J.R. Birge and F. Louveaux, *Introduction to stochastic programming*. Springer-Verlag (1997).

[8] G. Calafiore and M.C. Campi, Uncertain convex programs: randomized solutions and confidence levels. *Math. Program. A* **102** (2005) 25–46.

[9] G. Calafiore and L. El Ghaoui, Distributionally robust chance-constrained linear programs with applications. *J. Optim. Theor. Appl.* **130** (2006) 1–22.

[10] A. Charnes and W.W. Cooper, Chance-constrained programming. *Manage. Sci.* **5** (1959) 73–79.

[11] X. Chen, M. Sim and P. Sun, A Robust optimization perspective of stochastic programming. *Oper. Res.* (2007) **55** 1058–1071.

[12] G.B. Dantzig, Linear programming under uncertainty. *Manage. Sci.* **1** (1955) 179–206.

[13] E. Erdoğan and G. Iyengar, Ambiguous chance constrained problems and robust optimization. *Math. Program. Ser. B* **55** (20057) 37–61.

[14] C. Grinstead and J. Snell, *Introduction to probability*. American Mathematical Society, Providence, Rhode Island (1994).

[15] W.K. Klein Haneveld and M.H. van der Vlerk, Stochastic integer programming: state of the art (1998), available at http://citeseer.ist.psu.edu/kleinhaneveld98stochastic.html.

[16] W. Hoeffding, Probability inequalities for sums of bounded random variables. *Am. Stat. Assoc. J.* **58** (1963) 13–30.

[17] P. Kall and S.W. Wallace, *Stochastic programming*. Wiley, Chichester (1994).

[18] O. Klopfenstein and D. Nace, A robust approach to the chance-constrained knapsack problem. to appear in *Oper. Res. Lett.*

[19] O. Klopfenstein, Tractable algorithms for chance-constrained combinatorial problems, http://perso.rd.francetelecom.fr/klopfenstein/Papers/rmilp_online.pdf (long version of the current paper).

[20] A.M. Mood and F.A. Graybill, *Introduction to the theory of statistics*. McGraw-Hill Book Company Inc., New York (1963).

[21] A. Nemirovski and A. Shapiro, Convex Approximations of chance constrained programs. *SIAM J. Optim.* **17-4** (2006) 969–996.

[22] A. Nemirovski and A. Shapiro, Scenario approximations of chance constraints, in *Probabilistic and Randomized Methods for Design under Uncertainty*, edited by G. Calafiore and F. Dabbene. Springer, London (2005) 3–48.

[23] Y. Nikulin, *Robustness in combinatorial optimization and scheduling theory: an annotated bibliography*, www.optimization-online.org/DB_FILE/2004/11/995.pdf (2004).

[24] A. Ruszczyński, Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. *Math. Program. A* **93** (2002) 195–215.

[25] N.V. Sahinidis, Optimization under uncertainty: state-of-the-art and opportunities. *Comput. Chem. Eng.* **28** (2004) 971–983.

[26] S.R. Tayur, R.R. Thomas and N.R. Natraj, An algebraic geometry algorithm for scheduling in the presence of setups and correlated demands. *Math. Program.* **69-3** (1995) 369–401.