Pattern-Based Modeling and Solution of Probabilistically Constrained Optimization Problems — Source link

Miguel A. Lejeune

Institutions: George Washington University

Published on: 01 Nov 2012 - Operations Research (INFORMS)

Topics: Stochastic optimization, Stochastic programming, Probability distribution, Random variable and Linear programming

OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0030-364X | EISSN 1526-5463 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Pattern-Based Modeling and Solution of Probabilistically Constrained Optimization Problems

Miguel A. Lejeune
George Washington University, Washington, DC, 20052, USA; mlejeune@gwu.edu

We propose a new modeling and solution method for probabilistically constrained optimization problems. The methodology is based on the integration of the stochastic programming and combinatorial pattern recognition fields. It permits the fast solution of stochastic optimization problems in which the random variables are represented by an extremely large number of scenarios. The method involves the binarization of the probability distribution, and the generation of a consistent partially defined Boolean function (pdBf) representing the combination (F, p) of the binarized probability distribution F and the enforced probability level p. We show that the pdBf representing (F, p) can be compactly extended as a disjunctive normal form (DNF). The DNF is a collection of combinatorial p-patterns, each of which defining sufficient conditions for a probabilistic constraint to hold. We propose two linear programming formulations for the generation of p-patterns which can be subsequently used to derive a linear programming inner approximation of the original stochastic problem. A formulation allowing for the concurrent generation of a p-pattern and the solution of the deterministic equivalent of the stochastic problem is also proposed. Results show that large-scale stochastic problems, in which up to 50,000 scenarios are used to describe the stochastic variables, can be consistently solved to optimality within a few seconds.

Subject classifications: Programming: stochastic; Probability; Combinatorial Pattern; Probabilistic Constraint; Boolean Programming
History: Accepted, September 2012

1. Problem Formulation, Literature Review, and Contributions

In this paper, we propose a new modeling and numerical solution framework for stochastic programming problems (Prékopa, 1995; Birge and Louveaux, 1997; Ruszczyński and Shapiro, 2003). The methodology is based on pattern recognition (Vapnik, 1998; Grenander and Miller, 2007) and, in particular, on the derivation of logical and combinatorial patterns (Hammer, 1986; Martinez-Trinidad and Guzmán-Arenas, 2001; Truemper, 2004; Triantaphyllou and Felici, 2006). The framework allows for the deterministic reformulation and solution of probabilistically constrained programming problems of the form:

$$\begin{aligned} \min\ & q^{\mathsf{T}} x \\ \text{subject to } & Ax \geq b \\ & \mathbb{P}(h_j x \geq \xi_j,\ j \in J) \geq p \\ & x \geq 0 \end{aligned} \quad . \tag{1}$$

The notation $|J|$ refers to the cardinality of the set J, ξ is a $|J|$-dimensional random vector which has a multivariate probability distribution with finite support, x is the m-dimensional vector of decision variables, $q \in \mathcal{R}^m, b \in \mathcal{R}^d$, $A \in \mathcal{R}^{d\times m}$ and $h \in \mathcal{R}^{|J|\times m}$ are deterministic parameters, p is a prescribed probability or reliability level, and the symbol $\mathbb{P}$ refers to a probability measure. We consider the most general and challenging case in which there is no independence restriction between the components ξ_j of ξ. Thus,

$$\mathbb{P}(h_j x \geq \xi_j,\ j \in J) \geq p \tag{2}$$

is a joint probabilistic constraint which enforces that the combined fulfillment of a system of $|J|$ linear inequalities $\sum_{k=1}^{m} h_{jk}x_k \geq \xi_j$ must hold with a $|J|$-variate joint probability. Stochastic programming problems of this form are non-convex and very complex to solve.

1.1. Literature Review

Programming under probabilistic constraints has been extensively studied (see Prékopa (2003) for a review), and has been used for many different purposes ranging from the replenishment process in military operations (Kress et al., 2007), the enforcement of cycle service levels in a multi-stage supply chain (Lejeune and Ruszczyński, 2007), the construction of pension funds (Henrion, 2004), the monitoring of pollution level (Gren, 2008), etc. Probabilistic constraints with a random right-hand side have a deterministic technology matrix H in (1), while the stochastic component is in the right-hand side of the inequality $Hx \geq \xi$ subject to the probabilistic requirement. Stochastic optimization problems with individual (Charnes et al., 1958) probabilistic constraints (i.e., ξ is a one-dimensional vector) have a deterministic equivalent, whose continuous relaxation is straightforward to derive using the quantile of the one-dimensional random variable. However, the modeling of the reliability of a system through a set of individual probabilistic constraints does not allow the attainment of a system-wide reliability level (Prékopa, 1995), but instead enforces a certain reliability level for each individual part of the system. To that end, joint probabilistic constraints, first analyzed by Miller and Wagner (1965) under the assumption of independence between each component of the random vector, are needed. Prékopa (1970) studied the most general setting by removing the independence assumption between the components of the system.

A key factor for the computational tractability of stochastic problems with joint probabilistic constraints concerns the convexity property of the feasible set. Prékopa (1973) showed that, if the functions $h_j x - \xi_j \geq 0$ are concave in x and ξ, and ξ is continuously distributed with logarithmically concave probability density function, then the set of vectors x satisfying the joint probabilistic constraint is convex, allowing therefore to resort to a solution method based on convex programming techniques. However, such convexity properties do not apply when the random variables are discretely distributed. The corresponding optimization problems are non-convex and NP-hard, and have been receiving particular attention lately (Dentcheva et al., 2001; Ruszczyński, 2002; Cheon et al., 2006; Lejeune and Ruszczyński, 2007; Luedtke and Ahmed, 2008; Lejeune and Noyan, 2010; Luedtke et al., 2010, Saxena et al., 2010; Tanner and Ntaimo, 2010; Dentcheva and Martinez, 2012; Küçükyavuz, 2012; Lejeune, 2012).

Three main families of solution approaches for the above probabilistically constrained optimization problems can be found in the literature. The first one relies on the concept of p-efficiency (Prékopa, 1990), which requires the a priori uncovering of the finite set of p-efficient points (pLEPs) and permits the derivation of a mixed-integer programming (MIP) or disjunctive problem equivalent to the stochastic one. The reformulated problem can be solved through a convexification process and the cone generation algorithm (Dentcheva et al., 2001), with a primal-dual algorithm (Dentcheva et al., 2004), with a specialized column generation algorithm (Lejeune and Ruszczyński, 2007), or with an augmented Lagrangian method (Dentcheva and Martinez, 2012). While pLEPs are typically generated with enumerative schemes (see, e.g., Prékopa et al., 1998; Beraldi and Ruszczyński, 2002; Prékopa, 2003), Lejeune and Noyan (2010) and Lejeune (2012) propose MIP formulations to identify pLEPs. To solve the MIP problems, Lejeune and Noyan (2010) develop a modular method based on a bundle preprocessing algorithm and an outer approximation method. This latter solves a series of increasingly tighter outer approximations by using a set of valid strengthening inequalities and a fixing strategy. Lejeune (2012) revisits the p-efficiency concept and introduces the concept of an e^p-pattern. An e^p-pattern defines a pLEP as a conjunction of literals and is the Boolean Programming representation of a pLEP. Lejeune (2012) shows that the exhaustive collection of pLEPs can be represented as an irreducible disjunctive normal form that can be derived in an integrated (i.e., solution of a large-scale MIP) or sequential (i.e., solution of a finite series of MIPs) fashion.

The second family of solution methods associates a binary variable with each possible realization of the random vector and, then, substitutes an MIP problem of very large dimensionality (i.e., one binary variable per possible realization) for the original stochastic one (Cheon et al., 2006). To solve the resulting MIP problem, which contains a cover and "big-M" constraints, Ruszczyński (2002) develop specialized cutting planes which he embeds in a branch-and-cut algorithm. Cheon et al. (2006) design a branch-reduce-cut algorithm that iteratively partitions the feasible region and uses bounds to fathom inferior partitions. Luedtke et al. (2010) propose stronger MIP formulations for which they generate a family of valid inequalities, which are subsumed by the facet-defining family of cuts derived by Küçükyavuz (2012). In a set of recent studies, a sample approximation problem (Luedtke and Ahmed, 2008) is used to generate feasible solutions and optimality bounds for problems with joint probabilistic constraints. Tanner and Ntaimo (2010) propose an MIP formulation which they strengthen by deriving irreducibly infeasible subsystem optimality cuts. It was also shown that MIP reformulations of the probabilistic set covering problem can be solved in a very computationally efficient way (Saxena et al., 2010).

The third type of approaches consists in deriving safe approximations (Calafiore and Campi, 2005; Nemirowski and Shapiro, 2006) that take the form of convex optimization problems whose optimal solution is not always close to the true optimal solution. In fact, the probability level enforced by these techniques can be much larger than the prescribed level p. If the decision-maker is willing to trade some safety level for lower costs, and sets accordingly the reliability level p to moderately high values (say $p = 0.9, 0.95$), then the robust approximation might not always be suitable (Luedtke and Ahmed, 2008).

1.2. Motivation and Contributions

The fundamental contribution of this paper resides in the development of a novel solution methodology for stochastic programming problems. To the best of our knowledge, this is the first time that techniques from the pattern recognition field (Fukunaga, 1990; Vapnik, 1998; Duda and Stork, 2001; Grenander and Miller, 2007) are employed for the optimization of probabilistically constrained problems. Pattern recognition has been primarily used for feature selection, unsupervised classification, clustering, data mining or image processing purposes. The expected outcomes of pattern-based methods differ depending on whether they are used for classification or for optimization. With classification objectives in mind, logical / combinatorial pattern methods (Hammer, 1986; Boros et al., 1997; Djukova and Zhuravlev, 2000; Ruiz-Shulcloper and Abidi, 2002; Truemper, 2004; Triantaphyllou and Felici, 2006) are used to derive ”rules” that separate data points belonging to different categories. In the stochastic optimization context of this paper, the extracted patterns provide a compact representation of sets of conditions that are sufficient for the satisfaction of a probabilistic constraint, and can be used to derive deterministic reformulations of the stochastic problem. Besides its novelty, a crucial aspect of the proposed framework is that it allows the fast exact solution of stochastic optimization problems in which the random variables are represented by an extremely large number of scenarios. We describe below the main elements of the proposed methodology and discuss the organization of the paper.

In Section 2, we introduce the concept of a cut point, define a binarization method for a probability distribution, propose a method for selecting relevant realizations, and represent the combination (F, p) of the binarized probability distribution F of the random variable ξ and the enforced probability level p as a partially defined Boolean function. In Section 3, we extend the pdBf representing (F, p) as a disjunctive normal form (DNF), which is a collection of combinatorial p-patterns. Each of those defines sufficient conditions for the probabilistic constraint (2) to hold. Then, we propose a new mathematical programming method for the derivation of combinatorial p-patterns. Two integer programming and two linear programming formulations are presented. An interesting contribution of the method is that it offers a remedy to an issue associated with enumerative methods, which are highly efficient for the generation of patterns of small degrees, but turn out to be not as well performing when large-degree patterns need to be extracted (Boros et al., 2000). In Section 4, we show how we can use the combinatorial patterns to derive a linear programming inner approximation and a mixed-integer programming deterministic equivalent of the probabilistically constrained problem (1). Section 5 discusses the numerical implementation of the proposed methodology. Section 6 provides concluding remarks.

2. Representation of (F, p) as a Partially Defined Boolean Function

In this section, we shall first discuss the binarization process of the probability distribution and show how this allows the representation of the combination (F, p) of the probability distribution F and the prescribed probability level p as a partially defined Boolean function (pdBf). We shall then present the required properties of the set of cut points used for the binarization process and define the set of relevant realizations considered for the pattern generation process.

2.1. Binarization of Probability Distributions

We develop an approach to binarize probability distributions with finite support. We denote by Ω the set of possible realizations $k \in \Omega$ of the $|J|$-dimensional random vector ξ with cumulative distribution function F. Each realization k is represented by the $|J|$-dimensional deterministic vector $\omega^k = \left[\omega_1^k, \ldots, \omega_{|J|}^k\right]$: $\mathbb{P}(\xi \leq \omega^k) = F(\omega^k)$. The marginal probability distributions are denoted by $\mathbb{P}(\xi_j \leq \omega_j^k) = F_j(\omega_j^k)$, $j \in J$. The example below is used throughout the manuscript to illustrate our approach.

Example 1 *Consider the probabilistically constrained problem*

$$\begin{aligned} &\min\ x_1 + 2x_2 \\ &\textit{subject to}\ \mathbb{P}\left\{ \begin{array}{l} 8 - x_1 - 2x_2 \geq \xi_1 \\ 8x_1 + 6x_2 \geq \xi_2 \end{array} \right\} \geq 0.7 \\ &x_1, x_2 \geq 0 \end{aligned} \tag{3}$$

where the random vector $\xi = [\xi_1, \xi_2]$ accepts ten equally likely realizations k represented by $\omega^k = [\omega_1^k, \omega_2^k]$ and has the following bivariate probability distribution:

Table 1 Probability Distribution

k	ω_1^k	ω_2^k	$F_1(\omega_1^k)$	$F_2(\omega_2^k)$	$F(\omega^k)$
1	6	3	1	0.2	0.2
2	2	3	0.3	0.2	0.1
3	1	4	0.2	0.3	0.1
4	4	5	0.7	0.4	0.3
5	3	6	0.4	0.5	0.3
6	4	8	0.7	0.7	0.5
7	6	8	1	0.7	0.7
8	1	9	0.2	0.9	0.2
9	4	9	0.7	0.9	0.7
10	5	10	0.8	1	0.8

The feasibility set defined by the probabilistic constraint is non-convex. It is the union of the polyhedra $\{(x_1, x_2) \in \mathcal{R}_+^2 : 8 - x_1 - 2x_2 \geq 6,\ 8x_1 + 6x_2 \geq 8\}$ and $\{(x_1, x_2) \in \mathcal{R}_+^2 : 8 - x_1 - 2x_2 \geq 4,\ 8x_1 + 6x_2 \geq 9\}$.

We first introduce the concepts of p-sufficient and p-insufficient realizations.

Definition 1 *A realization k is called p-***sufficient** *if and only if $F(\omega^k) \geq p$ and is p-***insufficient** *if $F(\omega^k) < p$.*

The inequality sign in $\xi \leq \omega^k$ must be understood componentwise. We obtain a partition of the set Ω of realizations into two disjoint sets of p-sufficient Ω^+ and p-insufficient Ω^- realizations such that: $\Omega = \Omega^+ \bigcup \Omega^-$ with $\Omega^+ \bigcap \Omega^- = \emptyset$. Using the concept of cut point, we shall now binarize the probability distribution and obtain the binary projection Ω_B of Ω. We denote by $n = \sum_{j \in J} n_j$ the sum of the number n_j of cut points for each component ξ_j.

Definition 2 *The* **binarization process** *is the mapping $\mathbb{R}^{|J|} \to \{0,1\}^n$ of ω^k into an n-binary vector*

$$\beta^k = \left[\beta^k_{11}, \ldots, \beta^k_{n_1 1}, \ldots, \beta^k_{ij}, \ldots, \beta^k_{n_j j}, \ldots\right] , \tag{4}$$

such that the value of each component β^k_{ij} is defined with respect to a **cut point** *c_{ij} as follows:*

$$\beta^k_{ij} = \begin{cases} 1 & if\ \omega^k_j \geq c_{ij} \\ 0 & otherwise \end{cases} , \tag{5}$$

where c_{ij} denotes the i^{th} cut point associated with component ξ_j,

$$i' < i \;\Rightarrow\; c_{i'j} < c_{ij} \;,\; i = 1, \ldots, n_j, j \in J\;. \tag{6}$$

The notation β_{ij} refers to the i^{th} binary attribute associated with component ξ_j. The set of cut points is used to generate a binary image β^k of each realization initially represented by the numerical vector ω^k. Each point β^k is a vertex of the n-dimensional unit cube $\{0,1\}^n$, and its component β^k_{ij} takes value 1 (resp., 0) if the value ω^k_j taken by ξ_j in realization k is larger than or equal to (resp., strictly smaller than) the cut point c_{ij}. As an illustration, we consider the set of cut points

$$C = \{c_{11} = 4;\ c_{21} = 5;\ c_{31} = 6;\ c_{12} = 8;\ c_{22} = 9;\ c_{32} = 10\} \tag{7}$$

to binarize the numerical components ω_1 and ω_2. The set (7) includes three cut points defined with respect to each component ($n_1 = n_2 = 3$ and $n = 6$). The central part of Table 2 displays the binarization of the probability distribution of ξ (see Example 1) with the set of cut points (7).

Note that the binarization process arranges the cut points in ascending order (6) and defines a set of regularized Boolean vectors (Crama, Hammer, 2011).

Lemma 3 *The binarization process described in Definition 2 generates a* **regularized** *set of Boolean vectors, i.e., for every group $\left(\beta_{1j}, \ldots, \beta_{n_j,j}\right)$, $j \in J$ of Boolean variables, we have*

$$\beta^k_{ij} \leq \beta^k_{i'j} \;,\; j \in J \;,\; k \in \Omega \quad if \quad i' < i\;. \tag{8}$$

The binarization process defines the binary projection Ω_B of Ω: $\Omega_B = \Omega_B^+ \bigcup \Omega_B^- \subseteq \{0,1\}^n$, where Ω_B^+ and Ω_B^- respectively denote the sets of binarized p-sufficient and p-insufficient realizations. This permits the representation of the combination (F, p) of a probability distribution F and a probability level p as a partially defined Boolean function (pdBf).

Definition 4 (Eiter et al., 2002) *A* **partially defined Boolean function** *defined by the pair of disjoint sets* $\Omega_B^+, \Omega_B^- \subseteq \{0,1\}^n$ *is a mapping* $g : (\Omega_B^+ \bigcup \Omega_B^-) \to \{0,1\}$ *such that* $g(k) = 1$ *if* $k \in \Omega_B^+$ *and* $g(k) = 0$ *if* $k \in \Omega_B^-$.

Although the domain of the mapping g is in $\{0,1\}^n$, we use an index k (resp., $g(k)$) to denote an n-dimensional point (resp., function value) for the sake of notational convenience. We shall thereafter denote by $g(\Omega_B^+, \Omega_B^-)$ a pdBf defined by the pair of sets Ω_B^+, Ω_B^-. The right-hand side of Table 2 displays the truth table of the pdBf obtained with the set of cut points (7) for $p = 0.7$.

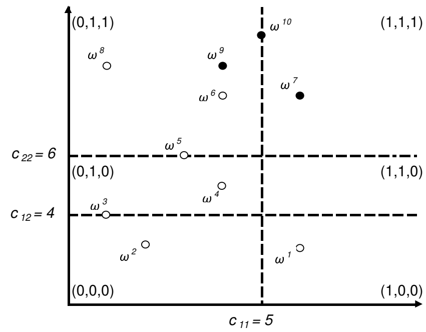
Table 2 Realizations, Binary Images, and Truth Table of Partially Defined Boolean Function

k	Numerical Representations		Truth Table of Partially Defined Boolean Function							
			Binarized Images						$g(k)$	
	ω_1^k	ω_2^k	β_{11}^k	β_{21}^k	β_{31}^k	β_{12}^k	β_{22}^k	β_{32}^k		
1	6	3	1	1	1	0	0	0	0	Set Ω_B^- of p-insufficient realizations
2	2	3	0	0	0	0	0	0	0	
3	1	4	0	0	0	0	0	0	0	
4	4	5	1	0	0	0	0	0	0	
5	3	6	0	0	0	0	0	0	0	
6	4	8	1	0	0	1	0	0	0	
8	1	9	0	0	0	1	1	0	0	
7	6	8	1	1	1	1	0	0	1	Set Ω_B^+ of p-sufficient realizations
9	4	9	1	0	0	1	1	0	1	
10	5	10	1	1	0	1	1	1	1	

2.2. Properties of Set of Cut Points

In Example 1, the binarization of the probability distribution with respect to the six cut points in (7) yields a pdBf such that the sets Ω_B^+ and Ω_B^- do not intersect. However, not all sets of cut points allow this. Consider, for example, the set of cut points $\{c_{11} = 5;\ c_{12} = 4;\ c_{22} = 6\}$ that generates the same binary image $(0, 1, 1)$ (Figure 1) for the p-sufficient realization 9 and the p-insufficient ones 5, 6 and 8. Such a set of cut points does not preserve the disjointedness between the sets of p-sufficient and p-insufficient realizations. Indeed, it results in p-sufficient and p-insufficient realizations having the same binary projection and impedes the derivation of the conditions that are necessary for (2) to hold. Clearly, the ability to accurately separate p-sufficient from p-insufficient realizations is a prerequisite for the reformulation to the stochastic problem (1). This requires the generation of a consistent set of cut points.

Figure 1 Inconsistent Set of Cut Points



Definition 5 (Boros et al., 1997) *A set of cut points is* **consistent** *if the sets* Ω_B^+ *and* Ω_B^- *associated with the pdBf* $g(\Omega_B^+, \Omega_B^-)$ *are disjoint. If this is the case,* $g(\Omega_B^+, \Omega_B^-)$ *is a consistent pdBf.*

We introduce the concept of sufficient-equivalent set of cut points. It is the cornerstone for the construction of a consistent pdBf representing the probability distribution F and the probability level p. Section 4 shows the importance of this concept for the derivation of inner approximations and an MIP reformulation equivalent to the stochastic problem (1) and containing a small number of binary variables.

Definition 6 *A* **sufficient-equivalent** *set of cut points* C^e *comprises a cut point* c_{ij} *for any value* ω_j^k *taken by any of the p-sufficient realizations on any component j:*

$$C^e = \bigcup_{j=1}^{|J|} C_j \,, \quad \textit{where} \quad C_j = \{\omega_j^k : k \in \Omega^+\} \,. \tag{9}$$

The pdBf $g(\Omega_B^+, \Omega_B^-)$ associated with the sufficient-equivalent set of cut points is called sufficient-equivalent pdBf. Proposition 7 is obvious and a direct consequence of Definition 6.

Proposition 7 *A sufficient-equivalent set of cut points is consistent.*

The construction of the sufficient-equivalent set of cut points is immediate. In our example, the sufficient-equivalent set of cut points is the one defined in (7): $C_1 = \{4, 5, 6\}$ and $C_2 = \{8, 9, 10\}$. Note that the combinatorial pattern literature (Boros et al., 1997; Hammer and Bonates, 2006; Ibaraki, 2011) describes several techniques (polynomial-time algorithm, set covering formulation) to build consistent set of cut points with special features (master or minimal set of cut points).

2.3. Set of Relevant Realizations

The objective is to derive a combinatorial pattern that defines sufficient conditions, possibly the minimal ones, for the probabilistic constraint (2) to be satisfied. In order to do so, we must not only take into consideration the realizations $k \in \Omega$ of the random vector, but we should also consider all points that could be p-sufficient. For k to be p-sufficient (i.e., $F(\omega^k \geq p)$), the $|J|$ following conditions must hold:

$$F_j(\omega_j^k) \geq p, j = 1, \ldots, |J| \,, \tag{10}$$

where F_j is the marginal probability distribution of ξ_j. Thus, for every j, we create the set Z_j

$$Z_j = \left\{\omega_j^k : F_j(\omega_j^k) \geq p, k \in \Omega\right\}, \; j = 1, \ldots, |J| \tag{11}$$

which contains the values that must be considered to identify the sufficient conditions for (2) to be satisfied. Then, we define the direct product (Prékopa, 2003)

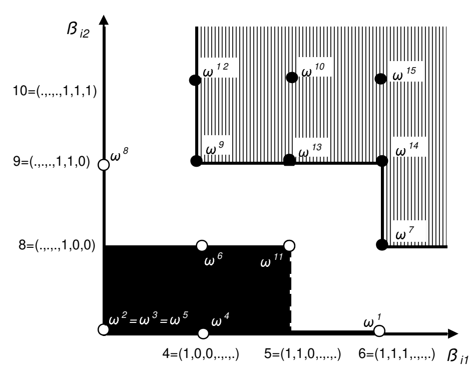
$$Z = Z_1 \times \ldots \times Z_j \times \ldots \times Z_{|J|} \,, \tag{12}$$

and obtain the **extended** set $\Omega \bigcup Z$ of realizations.

The application of the binarization process to the additional points included in Z provides their binarized images. In Example 1, the set Z comprises five realizations ($k = 11, \ldots, 15$ in Table 3).

Figure 2 shows that each p-sufficient realization is mapped into a binary vector which differs from all the binary vectors associated with p-insufficient realizations. The gray (resp., black) area in Figure 2 is the integer hull of the p-sufficient (resp., p-insufficient) realizations. All the points in the area between the two integer hulls correspond to vectors β with fractional values, which, by virtue of the binarization process (5), are numerical values that ξ_j cannot take and that do not belong to Z_j (11). The binarization process and the construction of the extended set of realizations enable the representation of the upper (resp., lower) envelope of the integer hull of the p-insufficient (resp., p-sufficient) points. Note that, if we do not consider realization 11 (which belongs to set Z) with binary image $\beta^{11} = (1, 1, 0, 1, 0, 0)$, we are not able to obtain the upper envelope of the integer hull of the p-insufficient points. This would be a problem for generating patterns defining sufficient conditions for the constraint (2) to hold.

Figure 2 Integer Hull of p-Sufficient and p-Insufficient Realizations



The binarization phase allows the elimination of a number of points from the extended set and the derivation of the set $\bar{\Omega}_B$ of **relevant** realizations. Several realizations have the same binary image (e.g., realizations 2 and 3) and we only include one of them in $\bar{\Omega}_B$. Recall that the objective is to derive patterns defining sufficient conditions for the satisfaction of (2). A well known set of necessary conditions for p-sufficiency is given by (10) which can be rewritten as

$$w_j^k \geq F_j^{-1}(p) \; = \; \min \left\{ \omega_j^k : \omega_j^k \in C_j \right\} \; = \; c_{1j}, \; j \in J \tag{13}$$

using the definition of the sufficient-equivalent set of cut points (9). The binarization process (5) implies further that:

$$w_j^k \geq c_{1j}, \; j \in J \; \Leftrightarrow \; \beta_{1j}^k = 1, \; j \in J \; . \tag{14}$$

A realization k such that $\beta_{1j}^k = 0$ for any $j \in J$ does not meet the basic preliminary condition and is a priori known to be p-insufficient. Thus, such a k is not needed to generate patterns separating p-sufficient realizations from p-insufficient ones and is not included in the set of relevant realizations. Table 3 gives the set of relevant realizations $\bar{\Omega}$ for Example 1. In order to simplify the narrative, we omit the adjective "relevant" in the remaining part of the manuscript, and simply refer to $k \in \bar{\Omega}_B^+$ (resp., $k \in \bar{\Omega}_B^-$) as a p-sufficient (resp. p-insufficient) realization.

Table 3 Set of Relevant Realizations $\bar{\Omega}_B$

	Numerical Representations		Binarized Images					
k	ω_1^k	ω_2^k	β_{11}^k	β_{21}^k	β_{31}^k	β_{12}^k	β_{22}^k	β_{32}^k
6	4	8	1	0	0	1	0	0
7	6	8	1	1	1	1	0	0
9	4	9	1	0	0	1	1	0
10	5	10	1	1	0	1	1	1
11	5	8	1	1	0	1	0	0
12	4	10	1	0	0	1	1	1
13	5	9	1	1	0	1	1	0
14	6	9	1	1	1	1	1	0
15	6	10	1	1	1	1	1	1

3. Mathematical Programming Approach for Combinatorial Patterns

In this section, we shall develop a mathematical programming approach allowing for the construction of combinatorial patterns that define sufficient conditions for the probabilistic constraint to hold. Prior to generating combinatorial patterns, we introduce the terminology and explain the rationale for the use of mathematical programming in pattern derivation.

3.1. Extension of the Partially Defined Boolean Function of (F, p)

Section 2 details how the binarization process permits the derivation of a pdBf that represents the combination (F, p) of the probability distribution F with the probability level p. The question that arises now is whether a compact extension (Urbano and Mueller, 1956) of the pdBf representing (F, p) can be derived.

Definition 8 (Eiter et al., 2002) *A function $f : \{0,1\}^n \to \{0,1\}$ is called an* **extension** *of a pdBf $g\left(\bar{\Omega}_B^+, \bar{\Omega}_B^-\right)$ if $\bar{\Omega}_B^+ \subseteq \bar{\Omega}_B^+(f) = \{k : f(k) = 1\}$ and $\bar{\Omega}_B^- \subseteq \bar{\Omega}_B^-(f) = \{k : f(k) = 0\}$.*

Boros et al. (1997) showed that a pdBf $g\left(\bar{\Omega}_B^+, \bar{\Omega}_B^-\right)$ has a Boolean extension if and only if $\bar{\Omega}_B^+ \bigcap \bar{\Omega}_B^- = \emptyset$, which is equivalent to saying that any consistent pdBf can be extended by a Boolean function. Thus, Proposition 7 implies that the sufficient-equivalent pdBf representing (F, p) can be extended as a Boolean function. With the existence of a Boolean extension for the pdBf ensured, the objective is to find an extension f that is defined on the same support set as $g(\bar{\Omega}_B^+, \bar{\Omega}_B^-)$ and that is as simple as possible. Since every Boolean function can be represented by a DNF, we shall extend $g(\bar{\Omega}_B^+, \bar{\Omega}_B^-)$ as a DNF which is a disjunction of a finite number of **combinatorial patterns**. Broadly defined, a combinatorial pattern is a logical rule that imposes upper and lower bounds on the values of a subset of the input variables.

3.2. Terminology

Before defining the DNF that extends $g(\bar{\Omega}_B^+, \bar{\Omega}_B^-)$, we introduce the key Boolean concepts and notations used in this paper and illustrate them with Example 1.

The Boolean variables $\beta_{ij}, i = 1, \ldots, n_j, j \in J$ and their negations or complements $\bar{\beta}_{ij}$ are called **literals**. A conjunction of literals $t = \bigwedge_{ij \in P} \beta_{ij} \bigwedge_{ij \in N} \bar{\beta}_{ij}$, $P \bigcap N = \emptyset$ constitutes a **term** whose **degree** d is the number $(|P| + |N| = d)$ of literals in it. The set P (resp., N) is the set of non-complemented (resp., complemented) literals involved in the definition of the term t. A disjunction $\bigvee_{s \in S} t_s$ of terms t_s is called a **disjunctive normal form** (DNF).

Definition 9 *A term t is said to* **cover** *a realization k, which is denoted by $t(k) = 1$, if the products of the values β_{ij}^k taken by k on the literals β_{ij} defining the term is equal to 1:*

$$t(k) = 1 \Leftrightarrow \bigwedge_{ij \in P} \beta_{ij}^k \bigwedge_{ij \in N} \bar{\beta}_{ij}^k = 1 \ .$$

The **coverage** of a term is the number of realizations covered by it. In our example, $t = \beta_{11}\, \bar{\beta}_{12}$ is a term of degree 2 covering the p-insufficient realizations 1 and 4, and $f = \beta_{11}\, \bar{\beta}_{12} \bigvee \beta_{31}\, \bar{\beta}_{32}$ is a DNF that contains two terms of degree 2: f covers two p-insufficient (1 and 4) and two sufficient (7 and 14) realizations.

It follows from Definition 8 that the DNF f extending the pdBf $g(\bar{\Omega}_B^+, \bar{\Omega}_B^-)$ must be such that each realization defined as p-sufficient (resp., p-insufficient) by the pdBf $g(\bar{\Omega}_B^+, \bar{\Omega}_B^-)$ must also be considered as p-sufficient (resp., p-insufficient) by the DNF f. This is equivalent to requiring that the DNF f covers all p-sufficient realizations and does not cover any p-insufficient ones: $f(k) = 1,\ k \in \bar{\Omega}_B^+$ and $f(k) = 0,\ k \in \bar{\Omega}_B^-$. The DNF $f = \bigvee_{s \in S} t_s$ includes a number $|S|$ of p**-patterns**, each defining sufficient conditions for (2) to hold. Evidently, for a probabilistic constraint to hold, one condition at least must be imposed on each component of the random vector ξ. More precisely, we need to have $\beta_{1j}^k = 1,\ j \in J$ (14) for k to be p-sufficient. This, and the fact that the regularization property implies that $\beta_{ij}^k \leq \beta_{i'j}^k, j \in J$ if $i' < i$ (8), mean that a p-pattern defining sufficient conditions for a probabilistic constraint to be satisfied must include at least one non-complemented literal β_{ij} associated with each component $j \in J$.

Definition 10 *A* p**-pattern** *is a term that covers at least one p-sufficient realization and does not cover any p-insufficient one*

$$\bigvee_{k \in \bar{\Omega}_B^+} t(k) = 1 \quad and \quad \bigwedge_{k \in \bar{\Omega}_B^-} t(k) = 0 \ ,$$

and includes at least one non-complemented literal β_{ij} *for each component* j *of the random vector* ξ.

Corollary 11 *The degree of a pattern defining sufficient conditions for (2) to hold is at least equal to* $|J|$.

A p-pattern is a combinatorial rule represented as a conjunction of literals. It is a subcube of the n-dimensional unit cube $\{0, 1\}^n$ that intersects $\bar{\Omega}_B^+$ (i.e., one or more p-sufficient realizations satisfy its conditions) but does not intersect $\bar{\Omega}_B^-$ (i.e., no p-insufficient realization satisfies its conditions). The term $\beta_{21}\ \beta_{32}$ is a p-pattern: it does not cover any p-insufficient realization, but covers the p-sufficient realizations 10 and 15. Finally, note that a p-pattern is easier to generate than and differs from an e^p-pattern that represents a p-efficient point (Lejeune, 2012).

3.3. Pattern Properties: Rationale for Mathematical Programming Generation

3.3.1. Properties In order to derive patterns that can be conveniently used for computational purposes, we shall attempt to derive prime patterns (Hammer et al., 2004).

Definition 12 *A pattern is* **prime** *if the removal of one of its literals transforms it into a term which is not a pattern.*

Basically, it means that a prime pattern does not include any redundant literals. We now investigate whether the pdBf representing (F, p) can take some particular functional form facilitating its computational handling. In particular, we consider the **monotonicity** property which, for Boolean functions, provides key computational advantages (Crama and Hammer, 2011).

Definition 13 (Radeanu, 1974) *A Boolean function f is positive (increasing) monotone, also called* **isotone***, if $x \leq y$ implies $f(x) \leq f(y)$.*

The inequality sign is understood componentwise. The conditions under which a pdBf can be extended as a positive Boolean function is given by Boros et al. (1997) as:

Lemma 14 *A pdBf $g\left(\bar{\Omega}_B^+, \bar{\Omega}_B^-\right)$ has a positive Boolean extension if and only if there is no $k \in \bar{\Omega}_B^+$ and $k' \in \bar{\Omega}_B^-$ such that $\beta^k \leq \beta^{k'}$.*

Lemma 14 is used to derive Theorem 15 which applies to the type of extension (i.e., extension of a pdBf representing the combination of a probability distribution and of a probability level) studied in this paper.

Theorem 15 *A consistent pdBf $g(\bar{\Omega}_B^+, \bar{\Omega}_B^-)$ representing (F, p) is extended as a positive Boolean function.*

Proof: *(i)* The binarization process (5) is defined in a way that if $\omega^{k'} \ngeq \omega^k$, then $\beta^{k'} \ngeq \beta^k$.
(ii) A cumulative probability distribution is positive monotone. Thus, if $F(\omega^{k'}) \leq F(\omega^k)$, then $\omega^{k'} \ngeq \omega^k$. Definition 1 states that $k \in \bar{\Omega}^+$ if and only if $F(\omega^k) \geq p$, and $k' \in \bar{\Omega}^-$ if and only if $F(\omega^{k'}) < p$. Thus, for every pair $(k, k'), k \in \bar{\Omega}^+, k' \in \bar{\Omega}^-$, we have $F(\omega^{k'}) < F(\omega^k)$ and $\omega^{k'} \ngeq \omega^k$.
Combining *(i)* and *(ii)*, we can see that for any pair $(k, k'), k \in \bar{\Omega}^+, k' \in \bar{\Omega}^-$, we have $\omega^{k'} \ngeq \omega^k$ implying $\beta^{k'} \ngeq \beta^k$. This, along with Lemma 14, completes the proof. □

It was shown (see Torvik and Triantaphyllou, 2009) that patterns included in a DNF that constitutes a positive Boolean function do not need to contain complemented literals. We denote by P_j the set of non-complemented literals associated with j and involved in the definition of a term t: $P = \bigcup_{j \in J} P_j$.

Lemma 16 *Consider a sufficient-equivalent set of cut points. Any term $t = \bigwedge_{ij \in P} \beta_{ij}$ with $P_j \neq \emptyset, j \in J$ that does not cover any p-insufficient realization is a p-pattern.*

Proof: It is enough to show that t necessarily covers at least one p-sufficient realization to prove that t is a p-pattern (Definition 10). Let $k' \in \bar{\Omega}^+$ be such that $\omega_j^{k'} = c_{n_j j}, j \in J$. The binarization process (5)-(6) implies that $\beta_{ij}^{k'} = 1, i = 1, \ldots, n_j, j \in J$. Obviously, $\prod_{ij \in P} \beta_{ij}^{k'} = 1$, and Definition 9 allows us to conclude that k' is covered by t, which is hence a p-pattern. □

Clearly, prime patterns (Definition 12) included in a DNF that is an isotone Boolean function do not contain complemented literals (Boros et al., 2000). For the problem at hand, this leads to the Lemma

Lemma 17 *Prime p-patterns do not contain any complemented literal $\bar{\beta}_{ij}$.*

which, combined with Proposition 11, indicates that

Lemma 18 *Prime p-patterns for realizations of a $|J|$-variate random variable are of degree $|J|$.*

Proof: Let $t = \bigwedge_{ij \in P} \beta_{ij} \bigwedge_{ij \in N} \bar{\beta}_{ij}$ be a p-pattern. From Lemma 17, we know that $N = \emptyset$ if t is a prime pattern. Consider that t includes two literals β_{ij} and $\beta_{i'j}$ associated with the same component j. Let $i' < i$ which implies that $c_{i'j} < c_{ij}$ (6) and $\beta_{ij} \leq \beta_{i'j}$ (8). If the removal of β_{ij} transforms t into a term that is not a pattern, then β_{ij} must be kept among the literals included in t. This makes $\beta_{i'j}$ redundant and the definition of a prime pattern requires its removal. On the other hand, if the removal of β_{ij} does not result in t covering any p-insufficient realization, then β_{ij} is not required and should be removed. This shows that prime p-patterns contain at most one literal per component j, and are thus of degree $|J|$. □

Lemma 19 follows immediately:

Lemma 19 *A pdBf $g(\bar{\Omega}^+_B, \bar{\Omega}^-_B)$ representing (F, p), where F is a $|J|$-variate probability distribution, can be extended as a DNF containing prime p-patterns of degree $|J|$ that do not include complemented literals.*

3.3.2. Rationale In the combinatorial data mining literature (Boros et al., 2000; Hammer and Bonates, 2006; Djukova et al., 2006), combinatorial patterns are usually generated by using term enumeration methods. Recent research related to the combinatorial methodology called logical analysis of data (Hammer, 1986) has led to major developments in this area and has shown that enumeration methods are very efficient (Alexe and Hammer, 2006; Hammer et al., 2006) when used for the generation of patterns of small degree (up to 4). The `LAD - Datascope 2.01` software package (Alexe, 2007) implements a variety of enumeration algorithms. However, enumerative techniques are extremely computationally expensive (Boros et al., 1997) when they are used to generate terms of larger degree. Indeed, the number of terms of degree up to d is equal to $\sum_{d'=1}^{d} 2^{d'} \binom{n}{d'}$ and increases very fast with the number n of Boolean variables (and cut points). This is a concern, since, as indicated by Lemma 18, prime p-patterns are of degree $|J|$, which is equal to the dimensionality of the multivariate probability distribution of ξ and potentially large. This motivates the development of a mathematical programming approach for the generation of patterns.

In combinatorial data mining, a set covering formulation has been proposed by Boros et al. (1997) for the generation of patterns. While patterns are derived to classify observations in data mining, the objective pursued in this paper is to use combinatorial patterns for the solution of probabilistically constrained optimization problems. Namely, the generated patterns permit the formulation of a tight linear programming inner approximation as well as that of the deterministic equivalent of probabilistically constrained problems. Besides the difference in objective, the mathematical programming formulations proposed in this paper substantially differ from those that can be found in the data mining literature. In particular, we propose two linear programming formulations for the derivation of patterns. The reader is referred to Jeroslow (1989), Hooker (2000, 2007), and Crama and Hammer (2011) for studies of the interplay between logic, Boolean mathematics, and optimization.

3.4. Mathematical Programming Derivation of p-Pattern

In this section, we propose four mathematical programming formulations for the generation of a p-pattern. Definition 10 shows that a p-pattern defines sufficient conditions for the probabilistic constraint (2) to hold. The optimal p-pattern is the one that enforces the minimal conditions for (2) to hold. However, it is not possible to identify some specific properties of an optimal p-pattern and to accordingly propose a tailored formulation for its generation. Thus, we shall focus on the derivation of a p-pattern that defines sufficient conditions that are “close to“ the minimal ones. The proposed formulations account for the following aspect. The optimal p-pattern, as well as those defining close-to-minimal conditions, represent faces of the lower envelope of the integer hull of the set of p-sufficient realizations, and are thus likely to have “large” coverage (see Figure 2).

3.4.1. Integer Programming Formulations The first integer programming (IP) formulation `IP1` is such that its optimal solution defines the p-pattern with maximal coverage. The following notations are used. The decision variables u_{ij} and y^k, respectively associated to the literals β_{ij} and to the p-sufficient realizations k, are binary (19)-(20). The value taken by u_{ij} defines the literals that are included in the p-pattern t: u_{ij} takes value 1 if β_{ij} is included in t, and is equal to 0 otherwise. The binary variable y^k identifies which p-sufficient realizations k are covered by t as defined by the solution of `IP1`: y^k takes value 1 if k is not covered by t, and can take value 0 otherwise.

The objective function (15) minimizes the number of p-sufficient realizations not covered by t. Each constraint in (16) forces y^k to take value 1 if the p-sufficient realization k is not covered by t. Each constraint in (17) does not permit t to cover any p-insufficient realization. Constraints (18) force the inclusion in t of one non-complemented literal (and no complemented literal) per j. We denote by $\mathbf{z}^*$ the optimal value of the objective function. Recall that the parameter β^k_{ij} indicates whether ω^k_j is at least equal to c_{ij} (5) and that we use a sufficient-equivalent set of n cut points.

Theorem 20 *Any feasible solution* $(\mathbf{u}, \mathbf{y})$ *of the integer programming problem* `IP1`

$$z = \min \sum_{k \in \bar{\Omega}^+_B} y^k \tag{15}$$

$$\textit{subject to} \quad \sum_{j \in J} \sum_{i=1}^{n_j} \beta^k_{ij} u_{ij} + |J| y^k \geq |J|, \quad k \in \bar{\Omega}^+_B \tag{16}$$

$$\sum_{j \in J} \sum_{i=1}^{n_j} \beta^k_{ij} u_{ij} \leq |J| - 1, \quad k \in \bar{\Omega}^-_B \tag{17}$$

$$\sum_{i=1}^{n_j} u_{ij} = 1, \qquad j \in J \tag{18}$$

$$u_{ij} \in \{0,1\}, \qquad j \in J, i = 1, \dots, n_j \tag{19}$$

$$y^k \in \{0,1\}, \qquad k \in \bar{\Omega}_B^+ \tag{20}$$

defines a prime p-pattern

$$t = \bigwedge_{\substack{u_{ij}=1 \\ j \in J, i=1,\dots,n_j}} \beta_{ij} \tag{21}$$

of degree $|J|$. Problem `IP1` *has an upper bound equal to $|\bar{\Omega}_B^+| - 1$ and its optimal solution $(\mathbf{u}^*, \mathbf{y}^*)$ defines the p-pattern with maximal coverage equal to $\left(\left|\bar{\Omega}_B^+\right| - \mathbf{z}^*\right)$.*

Proof: *(i)* Prime p-pattern: Let t (21) be a term defined by an arbitrary feasible solution $(\mathbf{u}, \mathbf{y})$ of `IP1` and $P = \{ij : u_{ij} = 1, j \in J, i = 1, \dots, n_j\}$ be the set of non-complemented literals in t.
It follows from Definition 9 that $k \in \bar{\Omega}_B$ is covered by t if and only if $\bigwedge_{ij \in P} \beta_{ij}^k = 1$, which is equivalent to

$$\sum_{ij \in P} \beta_{ij}^k = |P| = |J| \ . \tag{22}$$

Indeed, (18) and (19) ensure the inclusion of exactly one non-complemented literal per j in t. Thus, the number of literals in t is $|J| = |P|$.
Since $u_{ij} = 1, ij \in P$, and $u_{ij} = 0, ij \notin P$, implying that $\beta_{ij}^k \, u_{ij} = 0, ij \notin P$, then

$$\sum_{ij \in P} \beta_{ij}^k \ = \ \sum_{ij \in P} \beta_{ij}^k \, u_{ij} \ = \ \sum_{j \in J} \sum_{i=1}^{n_j} \beta_{ij}^k \, u_{ij} \ . \ . \tag{23}$$

Therefore, it follows that

$$\sum_{j \in J} \sum_{i=1}^{n_j} \beta_{ij}^k \, u_{ij} \leq |J| - 1, \ k \in \bar{\Omega}_B^- \quad \Rightarrow \quad \sum_{ij \in P} \beta_{ij}^k \leq |J| - 1, \ k \in \bar{\Omega}_B^- \ . \tag{24}$$

The above relationship implies that (17) prevents t from covering any $k \in \bar{\Omega}_B^-$ (see (22)). This, combined with Lemma 16, is enough to show that t is a p-pattern. As above-mentioned, (18) and (19) ensure that t is a p-pattern of degree $|J|$ and is therefore prime (Lemma 18).
(ii) Upper bound: We have shown above that if $k \in \bar{\Omega}_B$ is not covered by t, then $\sum_{j \in J} \sum_{i=1}^{n_j} \beta_{ij}^k \, u_{ij} \leq |J| - 1$. It follows that (16) forces $y^k, k \in \bar{\Omega}_B^+$ to take value 1 if $k \in \bar{\Omega}_B^+$ is not covered by t. Otherwise, y^k can take value 0. Part *(i)* of the proof indicates that any feasible solution of `IP1` determines a p-pattern which, by definition, covers at least one $k \in \bar{\Omega}_B^+$, for which $y^k = 0$. Therefore, the number of non-covered p-sufficient realizations $\sum_{k \in \bar{\Omega}_B^+} y^k$ is bounded from above by $|\bar{\Omega}_B^+| - 1$.
(iii) Coverage: The objective function maximizes the number $\sum_{k \in \bar{\Omega}_B^+} (1 - y^k)$ of $k \in \bar{\Omega}_B^+$ covered by t. Thus, the pattern defined by the optimal solution $(\mathbf{u}^*, \mathbf{y}^*)$ has maximal coverage equal to the difference between the number ($|\bar{\Omega}_B^+|$) of p-sufficient realizations and the number ($\mathbf{z}^* = \sum_{k \in \bar{\Omega}_B^+} y^k$) of those not covered by t. □

The number of binary variables in `IP1` is equal to $n + |\bar{\Omega}_B^+|$, and increases with the number of cut points and with the number of p-sufficient realizations. Note that `IP1` does not need to be solved to optimality, since any feasible solution defines a p-pattern, and that a pattern with maximal coverage is a strong pattern (Hammer et al., 2004).

Next, we formulate an MIP problem `IP2` which contains a significantly smaller number of binary variables than `IP1` and allows for the derivation of a p-pattern. The generated prime p-pattern t contains $|J|$ literals β_{ij}, and each literal defines a specific condition ($\omega_j^k \geq c_{ij}$) for a realization k to be covered by t. Instead of minimizing the number of p-sufficient realizations not covered by the pattern (see `IP1`), we shall now minimize the number of conditions imposed by the literals involved in t that are not satisfied by the p-sufficient realizations. If k is covered by t, then $\sum_{j \in J} \sum_{i=1}^{n_j} \beta_{ij}^k u_{ij} = |J|$ and constraint (26) requires $y^k = 0$. Otherwise, (26) forces y^k to be equal to the number $(|J| - \sum_{j \in J} \sum_{i=1}^{n_j} \beta_{ij}^k u_{ij})$ of conditions defined by the literals included in t that k does not satisfy. The resulting MIP problem `IP2` contains n binary variables instead of $(n + |\bar{\Omega}_B^+|)$ in `IP1`. The variables y^k are now continuous (27).

Theorem 21 *Any feasible solution* $(\mathbf{u}, \mathbf{y})$ *of the mixed-integer programming problem* `IP2`

$$z = \min \sum_{k \in \bar{\Omega}_B^+} y^k \tag{25}$$

$$\textit{subject to} \quad \sum_{j \in J} \sum_{i=1}^{n_j} \beta_{ij}^k u_{ij} + y^k = |J|, \quad k \in \bar{\Omega}_B^+ \tag{26}$$

$$0 \leq y^k \leq |J|, \qquad k \in \bar{\Omega}_B^+ \tag{27}$$

$$(17) - (19)$$

defines a prime p-pattern

$$t = \bigwedge_{\substack{u_{ij}=1 \\ j \in J, i=1,\ldots,n_j}} \beta_{ij}$$

of degree $|J|$ *and coverage* $|V|$ *with* $V = \{k : y^k = 0, k \in \bar{\Omega}_B^+\}$. *Problem* `IP2` *has an upper bound equal to* $|J| \cdot (|\bar{\Omega}_B^+| - 1)$.

Proof: *(i)* p-pattern: As shown in Theorem 20, constraints (17), (18) and (19) ensure that t is a prime p-pattern. Each constraint (26) allows y^k to take value 0 if and only if t covers $k \in \bar{\Omega}_B^+$. Thus, the coverage of t is equal to the cardinality of the set V.

(ii) Upper bound: Each y^k is bounded from above by $|J|$ (27). Any feasible solution of `IP2` defines a p-pattern (see *(i)*) that covers at least one $k \in \bar{\Omega}_B^+$. For $k \in \bar{\Omega}_B^+$ covered by t, y^k is equal to 0. The upper bound on the objective value of `IP2` is thus $|J| \cdot (|\bar{\Omega}_B^+| - 1)$. □

The `IP1` and `IP2` formulations for Example 1 are given below. The optimal solutions of `IP1` and `IP2` provide both the same p-pattern $t = \beta_{11}\beta_{22}$ with coverage equal to $|V| = 6$ (t does not cover realization 7), and $\mathbf{z}^*$ is equal to 1 for `IP1` and `IP2`.

`IP1`

$$
\begin{aligned}
z = \min\ & y^7 + y^9 + y^{10} + y^{12} + y^{13} + y^{14} + y^{15} \\
\text{s. to}\ & u_{11} + u_{21} + u_{31} + u_{12} + 2y^7 \geq 2 \\
& u_{11} + u_{12} + u_{22} + 2y^9 \geq 2 \\
& u_{11} + u_{21} + u_{12} + u_{22} + u_{32} + 2y^{10} \geq 2 \\
& u_{11} + u_{12} + u_{22} + u_{32} + 2y^{12} \geq 2 \\
& u_{11} + u_{21} + u_{12} + +u_{22} + 2y^{13} \geq 2 \\
& u_{11} + u_{21} + u_{31} + u_{12} + u_{22} + 2y^{14} \geq 2 \\
& u_{11} + u_{21} + u_{31} + u_{12} + u_{22} + u_{32} + 2y^{15} \geq 2 \\
& u_{11} + u_{12} \leq 1 \\
& u_{11} + u_{21} + u_{12} \leq 1 \\
& u_{11} + u_{21} + u_{31} = 1 \\
& u_{12} + u_{22} + u_{32} = 1 \\
& u_{ij} \in \{0, 1\},\ i = 1, 2, 3,\ \ j = 1, 2 \\
& y^k \in \{0, 1\},\ k = 7, 9, 10, 12, 13, 14, 15
\end{aligned}
$$

`IP2`

$$
\begin{aligned}
z = \min\ & y^7 + y^9 + y^{10} + y^{12} + y^{13} + y^{14} + y^{15} \\
\text{s. to}\ & u_{11} + u_{21} + u_{31} + u_{12} + y^7 = 2 \\
& u_{11} + u_{12} + u_{22} + y^9 = 2 \\
& u_{11} + u_{21} + u_{12} + u_{22} + u_{32} + y^{10} = 2 \\
& u_{11} + u_{12} + u_{22} + u_{32} + y^{12} = 2 \\
& u_{11} + u_{21} + u_{12} + +u_{22} + y^{13} = 2 \\
& u_{11} + u_{21} + u_{31} + u_{12} + u_{22} + y^{14} = 2 \\
& u_{11} + u_{21} + u_{31} + u_{12} + u_{22} + u_{32} + y^{15} = 2 \\
& u_{11} + u_{12} \leq 1 \\
& u_{11} + u_{21} + u_{12} \leq 1 \\
& u_{11} + u_{21} + u_{31} = 1 \\
& u_{12} + u_{22} + u_{32} = 1 \\
& u_{ij} \in \{0, 1\},\ i = 1, 2, 3,\ \ j = 1, 2 \\
& 0 \leq y^k \leq 2,\ k = 7, 9, 10, 12, 13, 14, 15
\end{aligned}
$$

3.4.2. Linear Programming Formulations In this section, we propose two linear programming formulations for the generation of p-patterns. This is another difference between the p-patterns and e^p-patterns whose generation requires the solution of an MIP problem.

Theorem 22 *Any feasible solution* $(\mathbf{u}, \mathbf{y})$ *of the linear programming problem* `LP1`

$$z = \min \sum_{k \in \bar{\Omega}_B^+} y^k \tag{28}$$

subject to $(16) - (18)$

$$0 \leq u_{ij} \leq 1, \quad j \in J, i = 1, \ldots, n_j \tag{29}$$

$$0 \leq y^k \leq 1, \quad k \in \bar{\Omega}_B^+ \tag{30}$$

defines a p-pattern

$$t \ = \bigwedge_{\substack{u_{ij} > 0 \\ j \in J, i = 1, \ldots, n_j}} \beta_{ij} \tag{31}$$

with coverage $|V|$ *with* $V = \{k : y^k = 0, k \in \bar{\Omega}_B^+\}$.

Proof: The presence of (17) ensures that t (31) is a p-pattern (Theorem 20). Every constraint (16) requires the associated k to be covered by t for y^k to take value 0. Thus, the coverage of t is $|V|$. □

Problem `LP1` is a linear program and is obviously simpler to solve than `IP1` and `IP2`. The "cost" of removing the integrality restrictions on the variables is twofold. First, although the objective function is still related to the coverage of the generated pattern, it cannot be interpreted anymore as representing the number of p-sufficient realizations covered by t (`IP1`) or as the number of conditions imposed by t that are not met by the p-sufficient realizations (`IP2`). Second, the pattern t defined by a feasible solution of `LP1` is not necessarily prime and can contain a number of literals larger than $|J|$, which could be inconvenient from a computational point of view. This can easily be remedied. Indeed, from the knowledge of the pattern t generated by `LP1`, one can immediately derive a prime p-pattern.

Corollary 23 *A prime p-pattern*

$$t = \bigwedge_{j \in J} \beta_{\bar{i}_j j} \tag{32}$$

with

$$\bar{i}_j = \arg\max_i u_{ij} > 0, \, j \in J \tag{33}$$

can immediately be derived from any feasible solution $(\mathbf{u}, \mathbf{y})$ *of the linear programming problem* `LP1`.

Proof: Let $(\mathbf{u}, \mathbf{y})$ be an arbitrary feasible solution for `LP1` and $P = \{ij : u_{ij} > 0, j \in J, i = 1, \ldots, n_j\}$. For k to be covered by t defined by (31), we need $\beta^k_{ij} = 1, ij \in P$. The regularization property (8) of the Boolean vector β implies that if $i' < i$, then $\beta^k_{i'j} \geq \beta^k_{ij}, \; j \in J$. Thus, all conditions $\beta^k_{ij} = 1, ij \in P$ associated with a component j can be subsumed by $\beta^k_{\bar{i}_j j} = 1, j \in J$ with $\bar{i}_j$ defined by (33): $\beta^k_{\bar{i}_j j} = 1$ implies that $\beta^k_{ij} = 1, i = 1, \ldots, \bar{i}_j$. Therefore, t defined by (32) includes only one literal $\beta_{\bar{i}_j j}$ per component j and defines the same conditions as t defined by (31): t defined by (32) is a prime p-pattern. □

In Example 1, the optimal solution for `LP1` allows the derivation of a prime p-pattern $t = \beta_{11}\beta_{22}$ with coverage equal to $|V| = 6$ and $\mathbf{z}^* = 0.5$.

In the second linear programming formulation `LP2`, we have a reduced set of $n + |J| + |\bar{\Omega}^-_B|$ constraints and only n continuous decision variables u. We introduce a set of parameters b_{ij} which can be viewed as the price of including the literal β_{ij} in the definition of the pattern t. The optimal solution $(\mathbf{u}^*)$ of `LP2` defines the "least costly" p-pattern. We propose the two following guidelines for the pricing approach and the determination of the weights b_{ij}:

- intra-component pricing: We differentiate the weights b_{ij} assigned to the literals associated with the same component j. The goal is to generate a p-pattern that defines the minimal (or close to minimal) conditions for the attainment of the probability level p. Accordingly, we want to include in the pattern t the literals imposing the least demanding conditions. Thus, for any given j and $i > i'$, it is preferable, when possible, to include $\beta_{i'j}$ than β_{ij} in t and we accordingly price $\beta_{i'j}$ cheaper than β_{ij} by setting $b_{ij} > b_{i'j}, j \in J$.
- inter-component pricing: The value of $b_{ij}, i = 1, \ldots n_j$ associated with component j is an increasing function of the cost associated with j in the objective function of the stochastic problem (1).

Theorem 24 *Any feasible solution* ($\mathbf{u}$) *of the linear programming problem* LP2

$$z = \min \sum_{j\in J} \sum_{i=1}^{n_j} b_{ij} u_{ij} \tag{34}$$
$$\textit{subject to} \quad (17)\ ;\ (18)\ ;\ (29)$$

defines a p-pattern

$$t \;=\; \bigwedge_{\substack{u_{ij}>0 \\ j\in J, i=1,\ldots,n_j}} \beta_{ij}\ .$$

The proof is the same as for LP1. As for LP1, a feasible solution for LP2 does not necessarily define a prime p-pattern, but we can apply Corollary 23 to construct a prime p-pattern.

In Example 1, we use the intra-component pricing approach and set $b_{ij} = i, i = 1, \ldots, n_j, j \in J$. The optimal solution of LP2 gives the p-pattern $t = \beta_{11}\beta_{22}$ with coverage equal to $|V| = 6$. The LP1 and LP2 formulations in Example 1 are given below.

LP1

$$\begin{aligned}
z = \ & \min\ y^7 + y^9 + y^{10} + y^{12} + y^{13} + y^{14} + y^{15} \\
\text{s. to}\ \ & u_{11} + u_{21} + u_{31} + u_{12} + 2y^7 \geq 2 \\
& u_{11} + u_{12} + u_{22} + 2y^9 \geq 2 \\
& u_{11} + u_{21} + u_{12} + u_{22} + u_{32} + 2y^{10} \geq 2 \\
& u_{11} + u_{12} + u_{22} + u_{32} + 2y^{12} \geq 2 \\
& u_{11} + u_{21} + u_{12} + +u_{22} + 2y^{13} \geq 2 \\
& u_{11} + u_{21} + u_{31} + u_{12} + u_{22} + 2y^{14} \geq 2 \\
& u_{11} + u_{21} + u_{31} + u_{12} + u_{22} + u_{32} + 2y^{15} \geq 2 \\
& u_{11} + u_{12} \leq 1 \\
& u_{11} + u_{21} + u_{12} \leq 1 \\
& u_{11} + u_{21} + u_{31} = 1 \\
& u_{12} + u_{22} + u_{32} = 1 \\
& 0 \leq u_{ij} \leq 1,\ i = 1,2,3,\ \ j = 1,2 \\
& 0 \leq y^k \leq 1,\ k = 7,9,10,12,13,14,15
\end{aligned}$$

LP2

$$\begin{aligned}
z = \ & \min\ b_{11}u_{11} + b_{21}u_{21} + b_{31}u_{31} \\
& \qquad + b_{12}u_{12} + b_{22}u_{22} + b_{32}u_{32} \\
\text{s. to}\ \ & u_{11} + u_{12} \leq 1 \\
& u_{11} + u_{21} + u_{12} \leq 1 \\
& u_{11} + u_{21} + u_{31} = 1 \\
& u_{12} + u_{22} + u_{32} = 1 \\
& 0 \leq u_{ij} \leq 1,\ i = 1,2,3,\ \ j = 1,2
\end{aligned}$$

In Section 5, we shall evaluate the numerical efficiency of the four proposed mathematical programming formulations and the time needed to solve them to optimality.

4. Linear Reformulation of Probabilistic Problems

4.1. Linear Programming Inner Approximation of Probabilistic Problems

In this section, we derive an inner approximation, taking the form of a **linear program**, for the probabilistically constrained problem (1). The construction of the inner approximation problem is based on the generation of a p-pattern using one of the formulations proposed in Section 3.4.

Theorem 25 *Consider a p-pattern $t = \bigwedge_{ij\in P} \beta_{ij}$, with P denoting the set of literals included in t. The linear programming problem* IALP

$$\begin{aligned} &\min q^{\mathsf{T}} x \\ &\textit{subject to } Ax \geq b \\ &\qquad h_j x \geq c_{ij},\ ij \in P \\ &\qquad x \geq 0 \end{aligned} \tag{35}$$

is an inner approximation of the probabilistic problem (1).

Proof: The definitions of a p-pattern and of its coverage (Definition 9 and 10) imply that, if $t = \bigwedge_{ij\in P} \beta_{ij}$ is a p-pattern, we have

$$\begin{cases} t(k) = 0,\ \forall k \in \bar{\Omega}_B^- & \Leftrightarrow \quad \beta_{ij}^k = 0, \text{ for at least one } ij \in P,\ \forall k \in \bar{\Omega}_B^- \\ t(k) = 1, \text{ for at least one } k \in \bar{\Omega}_B^+ & \Leftrightarrow \quad \beta_{ij}^k = 1,\ \forall\ ij \in P, \text{ for at least one } k \in \bar{\Omega}_B^+ \end{cases} . \tag{36}$$

The definition of the binarization process (5) permits us to rewrite the second relationship in (36) as:

$$\beta_{ij}^k = 1,\ \forall\ ij \in P \quad \Leftrightarrow \quad \omega_j^k \geq c_{ij},\ \forall ij \in P\ . \tag{37}$$

The construction of the sufficient-equivalent set of cut points (see (9) in Definition 6) implies that:

$$\omega_j^k = c_{ij},\ \ \forall ij \in P, \text{ for one } k \in \bar{\Omega}^+\ . \tag{38}$$

Let $k' \in \bar{\Omega}^+$ be the realization for which (38) holds:

$$\omega_j^{k'} = c_{ij},\ \ \forall ij \in P\ . \tag{39}$$

Since $k' \in \bar{\Omega}^+$, we have $\mathbb{P}\left(\omega^{k'} \geq \xi\right) \geq p$, and thus,

$$\omega_j^{k'} \leq h_j x,\ j \in J \quad \Rightarrow \quad \mathbb{P}(h_j x \geq \xi_j,\ j \in J) \geq p\ . \tag{40}$$

Using (39), we can now rewrite (40) as

$$c_{ij} \leq h_j x,\ ij \in P \quad \Rightarrow \quad \mathbb{P}(h_j x \geq \xi_j,\ j \in J) \geq p\ , \tag{41}$$

which was set out to prove. □

The above linear programming problem can be obtained by using any of the four formulations proposed for the generation of p-patterns. The key question that is addressed in Section 5.1 pertains to the tightness of the inner approximation obtained with the four proposed models. The tightness of an approximation problem is understood as the gap between the optimal value of the approximation problem and this of the stochastic problem (1).

4.2. Linear Deterministic Equivalent of Probabilistic Problems

We shall now derive a linear deterministic problem `DEIP` equivalent to the probabilistically constrained program (1). Problem `DEIP` takes the form of an MIP problem including a number of binary variables equal to the number of cut points used in the binarization process. The solution of `DEIP` allows for the concurrent *(i)* generation of the prime p-pattern defining the minimal conditions for the probabilistic constraint (2) to hold and *(ii)* reformulation and exact solution of the stochastic problem (1).

Theorem 26 *The mixed-integer programming problem* `DEIP`

$$\begin{aligned}
\min \quad & q^{\mathsf{T}} x \\
\textit{subject to} \quad & Ax \geq b \\
& h_j x \geq \sum_{i=1}^{n_j} c_{ij}\, u_{ij}\,, \qquad j \in J & (42)\\
& \sum_{j\in J}\sum_{i=1}^{n_j} \beta_{ij}^{k}\, u_{ij} \leq |J| - 1\,, \quad k \in \bar{\Omega}_B^{-} & (43)\\
& \sum_{i=1}^{n_j} u_{ij} = 1, \qquad j \in J & (44)\\
& u_{ij} \in \{0,1\}, \qquad j \in J, i = 1,\ldots,n_j & (45)\\
& x \geq 0
\end{aligned}$$

is a deterministic equivalent of the probabilistically constrained problem (1). The optimal solution $(\mathbf{u}^*, \mathbf{x}^*)$ *of* `DEIP` *gives the prime p-pattern*

$$t^* = \bigwedge_{ij\in P} \beta_{ij}\,, \quad \textit{with} \quad P = \left\{ij : u_{ij}^* = 1,\ j \in J, i = 1,\ldots,n_j\right\} \tag{46}$$

that defines the minimal conditions for the probabilistic constraint (2) to be satisfied.

Proof: Let $t = \bigwedge_{ij\in P} \beta_{ij}$ with $P = \{ij : u_{ij} = 1,\ j \in J, i = 1,\ldots,n_j\}$ be a term defined by an arbitrary feasible solution $(\mathbf{u}, \mathbf{x})$ of `DEIP`.

(i) Prime p-pattern: Constraints (43) do not allow t to cover any $k \in \bar{\Omega}_B^{-}$. Hence, t defined by an arbitrary feasible solution for `DEIP` is a p-pattern (Lemma 16), and (44) and (45) imply that t is prime, with degree $|J|$. Thus, t^* (46) is a prime p-pattern.

(ii) We show now that any feasible solution for `DEIP` is feasible for (1). The definition of a p-pattern (Definition 10) and the construction of the sufficient-equivalent set of cut points (Definition 6) imply that there exists $k' \in \bar{\Omega}^{+}$ covered by t such that

$$\omega_j^{k'} = c_{ij},\ ij \in P\,. \tag{47}$$

Constraints (44) and (45) ensure that exactly one term $u_{ij}, ij \in P$ in the left-hand side of each constraint (44) is non-zero and equal to one. Similarly, exactly one term $c_{ij}\, u_{ij}, ij \in P$ in the right-hand side of each constraint (42) is non-zero and equal to $c_{ij}, ij \in P$. This allows rewriting (47) as

$$\omega_j^{k'} = \sum_{i=1}^{n_j} c_{ij}\, u_{ij}, j \in J \,. \tag{48}$$

Since $k' \in \bar{\Omega}^+$, we obtain (40) which can be rewritten using (48) as

$$\sum_{i=1}^{n_j} c_{ij}\, u_{ij} \leq h_j x,\ ij \in P \quad \Rightarrow \quad \mathbb{P}\left(h_j x \geq \xi_j, j \in J\right) \geq p \,. \tag{49}$$

This shows that any feasible solution for `DEIP` is feasible for (1).

(iii) Next, we show that any feasible solution for (1) is feasible for `DEIP`. The points $k'' : \mathbb{P}(\xi \leq \omega^{k''}) \geq p$ are such (40) that

$$h_j x \geq \omega_j^{k''}, j \in J \Rightarrow \mathbb{P}(h_j x \geq \xi_j, j \in J) \geq p \,,$$

and form the feasible set defined by (2). For any such point k'', we can always find (by construction of the sufficient-equivalent set of cut points) a point $k' \in \bar{\Omega}^+$, such that $\omega^{k'} \leq \omega^{k''}$ and $\omega_j^{k'} = \bigvee_{i=1,\dots,n_j} c_{ij},\ j \in J$. Let $i'_j = \arg\max_i \beta_{ij}^{k'} = 1,\ j \in J$, and let's construct the vector u' defined by k':

$$u'_{ij} = \begin{cases} 1 & \text{if} \quad i = i'_j \\ 0 & \text{otherwise} \end{cases} \,. \tag{50}$$

To prove that any feasible solution for (1) is feasible for `DEIP`, we show that, for any $k'' : \mathbb{P}(\xi \leq \omega^{k''}) \geq p$, one can find $k' \in \bar{\Omega}^+ : \omega^{k'} \leq \omega^{k''}$ such that u' (50) is feasible for (42), (43), (44) and (45).

It is easy to see that u' is feasible for (44) and (45). As for the constraints (43), u' is feasible if

$$\sum_{j \in J} \sum_{i=1}^{n_j} \beta_{ij}^k\, u'_{ij} = \sum_{j \in J} \beta_{i'_j j}^k\, u'_{i'_j j} = \sum_{j \in J} \beta_{i'_j j}^k \leq |J| - 1,\ k \in \bar{\Omega}_B^- \,. \tag{51}$$

Clearly, the feasibility of the above constraints is ensured if

$$\beta_{i'_j j}^k = 0 \text{ for at least one } j,\ \forall k \in \bar{\Omega}_B^- \,. \tag{52}$$

It is shown in Theorem 15 that there is no $k \in \bar{\Omega}_B^-$ such that $\beta^k \geq \beta^{k'}$. This implies that

$$(\beta_{1j}^k, \dots, \beta_{n_j j}^k) < (\beta_{1j}^{k'}, \dots, \beta_{n_j j}^{k'}) \text{ for at least one } j,\ \forall k \in \bar{\Omega}_B^- \,.$$

Let $h \in J$ be a coordinate such that $(\beta_{1h}^k, \dots, \beta_{n_h h}^k) < (\beta_{1h}^{k'}, \dots, \beta_{n_h h}^{k'})$ for an arbitrary $k \in \bar{\Omega}_B^-$. Since $i'_h = \arg\max_i \beta_{ih}^{k'} = 1$, it follows that, for any $i > i'_h$, $\beta_{ih}^k = \beta_{ih}^{k'} = 0$. Thus, the vectors $\beta^{k'}$ and β^k differ only in terms of the i'_h first components, and we have

$$(\beta_{1h}^k, \dots, \beta_{i'_h h}^k) < (\beta_{1h}^{k'}, \dots, \beta_{i'_h h}^{k'}) \,. \tag{53}$$

The regularization property ((8) in Lemma 3) indicates that (53) can only be true if $\beta^k_{i'_h h} = 0 < \beta^{k'}_{i'_h h} = 1$. This shows that, for any $k \in \bar{\Omega}^-_B$, $\beta^k_{i'_j j} = 0$ for at least one j, which yields (52) and results in $\sum_{j\in J} \beta^k_{i'_j j}$ being bounded from above by $(|J|-1)$ for each $k \in \bar{\Omega}^-_B$ (51). This shows that u' is feasible for (43).

Concerning the feasibility of u' for (42), we must show that

$$h_j x \geq \sum_{i=1}^{n_j} c_{ij}\, u'_{ij},\, j \in J \;\Rightarrow\; \mathbb{P}(h_j x \geq \xi_j, j \in J) \geq p\ . \tag{54}$$

Since $i'_j = \arg\max_i \beta^{k'}_{ij} = 1$ with (5) setting $\beta^{k'}_{ij} = 1$ if $\omega^{k'}_j \geq c_{ij}$ and the fact that $\omega^{k'}_j = \bigvee_{i=1,\dots,n_j} c_{ij},\ j \in J$, we have $\omega^{k'}_j = c_{i'_j j}$. Further, the definition of u' (50) implies that

$$\omega^{k'}_j = c_{i'_j j} = \sum_{i=1}^{n_j} c_{ij}\, u'_{ij},\ j \in J\ . \tag{55}$$

Since $k' \in \bar{\Omega}^+$, we have (40). Using (55), we substitute $\sum_{i=1}^{n_j} c_{ij}\, u'_{ij}$ for $\omega^{k'}_j$ in (40) and obtain the desired result (54). We conclude that any feasible solution for (2) defines a solution u' feasible for (42), (43), (44) and (45). Thus, any feasible solution for (1) is feasible for `DEIP`.

Parts *(ii)* and *(iii)* show that the feasible sets of (1) and `DEIP` are equivalent, and the optimal solution of `DEIP` is thus optimal for (1). Part *(i)* shows that any feasible solution of `DEIP` defines a prime p-pattern. The optimal solution of `DEIP` defines the p-pattern imposing the minimal conditions for (2) to hold. □

Other MIP reformulation approaches have been proposed to derive a deterministic equivalent for (1). The MIP deterministic equivalent reformulations (Dentcheva et al., 2001; Prékopa, 2003; Lejeune, 2008) obtained by using the p-efficiency concept (Prékopa, 1990) associate one binary variable with each pLEP which must be found a priori. Luedtke et al. (2010) propose several MIP formulations in which a binary variable is associated with each scenario. In contrast to this, the number of binary variables in the proposed reformulation `DEIP` is not an increasing function of the number of scenarios used to describe the uncertain variables. It contains a significantly lower number (n) of binary variables, equal to the cardinality of the sufficient-equivalent set of cut points (Definition 6) used for the binarization process.

The deterministic equivalent formulation `DEIP` for Example 1 reads:

$$\begin{aligned} z = \ & \min\ x_1 + 2x_2 \\ \text{subject to}\ & u_{11} + u_{12} \leq 1 \\ & u_{11} + u_{21} + u_{12} \leq 1 \\ & 8 - x_1 - 2x_2 \geq 4u_{11} + 5u_{21} + 6u_{31} \\ & 8x_1 + 6x_2 \geq 8u_{12} + 9u_{22} + 10u_{32} \\ & u_{11} + u_{21} + u_{31} = 1 \end{aligned}$$

$$u_{12} + u_{22} + u_{32} = 1$$

$$u_{ij} \in \{0,1\},\ i = 1,2,3,\ \ j = 1,2$$

$$x_1, x_2 \geq 0$$

The optimal value is 1, the optimal solution is $(\mathbf{u}^*, \mathbf{x}^*) = (0,0,1,1,0,0,1,0)$, and the p-pattern defined by (46) is $t = \beta_{31}\,\beta_{12}$.

5. Numerical Implementation

This section evaluates the computational efficiency of the proposed combinatorial pattern approach. The first part compares the speed of the four mathematical programming methods for the generation of p-patterns and analyzes the tightness of the inner approximation obtained with these methods. The second part looks at the computational times needed to solve the deterministic equivalent reformulation of the probabilistic problem. The tests are conducted on hundreds of instances describing a supply chain problem and a cash matching problem.

In the supply chain problem, the set K of distributors must satisfy the random demand ξ of a set J of customers. The decision variables x_{kj} are the supply quantities delivered by a distributor k to a customer j. The model reads:

$$\begin{aligned}
\min \quad & \sum_{k\in K}\sum_{j\in J} q_{kj}x_{kj} && && (56)\\
\text{subject to} \quad & \sum_{j\in J} x_{kj} \leq M_k\,, && k \in K && (57)\\
& x_{kj} \leq V_{kj}\,, && k \in K, j \in J && (58)\\
& \mathbb{P}\left(\sum_{k\in K} x_{kj} \geq \xi_j, j \in J\right) \geq p && && (59)\\
& x \geq 0
\end{aligned}$$

The parameter q_{kj} represents the cost of supplying one unit from k to j. The objective function (56) minimizes the sum of the distribution costs. Constraints (57) upper-bound (M_k) the aggregated supply quantity delivered by k to all its customers. Constraints (58) upper-bound (V_{kj}) the individual supply quantity delivered by k to each customer j. Constraints (59) require that the distributors satisfy the demand of all of their customers with a large probability level p.

The parameters q_{kj}, M_k and V_{kj} of the above model were randomly generated from uniform distributions. The probability distribution of ξ is described with a finite set of Ω realizations defined as equally likely and sampled from a uniform distribution. We create 32 types of problem instances characterized by the tuple $(|J|, |\Omega|, p)$. The instances differ in terms of the dimension $(|J| = 10, 20)$ of the random vector, the number of realizations $(|\Omega| = 5000, 10000, 20000, 50000)$, and the enforced probability level

($p = 0.875, 0.9, 0.925, 0.95$). For each instance type, we generate five problem instances. Table 6 reports the time and gap averages over the five instances of each instance type.

Within the stochastic cash matching problem (Dentcheva et al., 2004; Henrion, 2004), a company builds a pension fund by investing its available capital in low-risk bonds. The optimization problem is of the form of (1). The goal of the allocation strategy is to maximize the amount of money available at the end of the planning horizon (typically 10 to 15 years), while generating, at each period, sufficient cash amounts for the company to cover its retirement allowances. The problem contains a joint probabilistic constraint which enforces that the cumulative cash inflow exceeds the cumulative random cash payments with probability p. The decision variables are the positions in each type of bonds. The principal and coupon payments of the bonds are known, while the retirement payments at each time-period are stochastic and follow a discrete distribution. The allocation strategy does not involve any rebalancing operation. The face value, the yield structure, and the maturity of the bonds were obtained from the CRSP [8] and Mergent [41] databases and were used to construct 32 types of problem instances characterized by the tuple $(|J|, |\Omega|, p)$. The instances differ in terms of the dimension ($|J| = 10, 15$) of the random vector, the number of realizations ($|\Omega| = 5000, 10000, 20000, 50000$), and the enforced probability level ($p = 0.875, 0.9, 0.925, 0.95$). For each instance type, we generate five problem instances. Table 7 reports the time and gap averages over the five instances of each type.

The binarization process is carried out with Matlab. The AMPL modeling language (Fourer et al., 2003) is used to formulate the mathematical programming problems which are solved with the CPLEX 11.1 solver. Each problem instance is solved on a 64-bit Dell Precision T5400 Workstation with Quad Core Xeon Processor X5460 3.16GHz CPU, and 4X2GB of RAM.

5.1. Pattern Generation and Solution of Inner Approximation

The fourth (resp., sixth, eighth, and tenth) column in Table 6 and Table 7 (see Online Appendix for the latter) reports, for each type of family (see the first three columns of Tables 6 and 7), the sum of the average computational times needed *(i)* to generate a pattern with the `IP1` (resp., `IP2`, `LP1`, and `LP2`) formulation and *(ii)* to solve the resulting linear programming inner approximation. It can be seen that the four approaches are very fast, even for problems in which the multivariate random variable is described by a large number of scenarios. The two linear programming formulations are obviously the fastest (at most 1.8 sec, and most often much less), but the IP formulations can also be solved quickly (at most 29 seconds for `IP1` and 5 seconds for `IP2`). It is not surprising to observe that the solution times for the `IP2` formulation are consistently smaller than those for `IP1`, since the former formulation contains a significantly lower number of binary variables. For the IP formulations, the average computational time is generally an increasing function of the dimension of the random vector and a decreasing function of the probability level p.

The next question to settle pertains to the tightness of the inner approximation derived from the patterns obtained with the four proposed formulations. We measure the tightness of the approximation by the relative optimality gap between the optimal value of the inner approximation and the optimal value of the original stochastic problem (1). For each type of instance, the fifth (resp., seventh, ninth, and eleventh) column in Tables 6 and 7 reports the average relative optimality gap of the inner approximation obtained by using the `IP1` (resp., `IP2`, `LP1`, and `LP2`) formulation. Table 4 reports the number M of instance types for which each formulation gives the smallest relative optimality gap. Besides being the fastest, the linear programming approach `LP2` is also the one that provides the tightest inner approximation for the largest number of instance types (21 for the supply chain problem and 19 for the cash matching problem). The `LP1` model provides the most conservative approximations.

Table 4 Tightness of Inner Approximation Approaches

	IP1	IP2	LP1	LP2	Problem
M	8	8	1	21	Supply Chain
	7	8	0	19	Cash Matching

5.2. Concurrent Pattern Generation and Solution of Deterministic Equivalent

For the 320 problem instances, we solve problem `DEIP` which allows for the simultaneous generation of the prime p-pattern defining the minimal conditions for the probabilistic constraint (2) to hold and the solution of the deterministic equivalent formulation of (1). Problem `DEIP` contains $|J|$ set partitioning constraints (44) which can be explicitly defined as special ordered set constraints of type one (SOS1).

The twelfth column in Table 6 shows that the deterministic equivalent formulation can be solved extremely fast (in at most 3 seconds) for each supply chain family instance. The number of integer variables in problem `DEIP` is equal to the number of cut points, which, everything else being equal, increases as the probability level decreases (see Definition 6). Thus, it is logical that the computing time increases as the value of p decreases. We observe that the computing time increases at a very moderate rhythm, which suggests that the method could be used for values of p even lower than those considered here. Similar conclusions can be drawn for the cash matching problem (see Table 7 in Online Appendix for the detailed results). For the cash matching instances, the longest time taken to solve `DEIP` to optimality is 2.125 seconds.

A key feature of the proposed approach is that the number of binary variables does not increase with the number of realizations. This is what allows the application of the method for cases in which the random variables are subject to a very fine discretization and are characterized by an extra large number of scenarios. The results attest that the solution time does not increase monotonically with the number of scenarios used to represent uncertainty. This is illustrated by Table 5 that displays the average, minimum, and maximum time to solve to optimality the deterministic equivalent problem **DEIP** for each considered number $|\Omega|$ of scenarios. The average computational time to solve the 50000-scenario instances is smaller than that for the 20000-scenario instances.

Table 5 Relationship between Solution Time for DEIP Problem and Number of Scenarios - Cash Matching Problem Instances (in seconds)

	$\lvert\Omega\rvert$			
	5000	10000	20000	50000
Average Time	0.215	0.268	0.542	0.470
Minimum Time	0.015	0.021	0.011	0.010
Maximum Time	1.064	1.125	2.125	2.035

To our knowledge, none of the existing methods has reported numerical results for problem instances in which the number of scenarios is larger than 3000 (e.g., up to 3000 in Luedtke et al., 2010 and 500 in Küçükyavuz, 2012). We have implemented the method proposed by Luedtke et al. (2010) and we could not obtain the optimal solution for any of the problem instances containing 20,000 or more scenarios in one hour of CPU time. The proposed method is thus an excellent alternative to the existing algorithmic techniques.

6. Conclusion

We propose a novel methodology to solve probabilistically constrained optimization problems by using concepts from the combinatorial pattern recognition field. Combinatorial patterns are able to capture the ”interactions” between the components of a multi-dimensional random vector and their impact on making it possible to reach a predefined reliability level. Patterns have the capability not only to identify the variables that individually influence the probability level p, but also to capture the collective effect of the values of those variables on the attainment of the prescribed probability level. Besides being new, the proposed method also scales well. It permits the very fast solution of stochastic optimization problems in which the random variables are represented by an unprecedently large number of scenarios.

The presented framework introduces the concept of a cut point and describes a binarization method for a probability distribution with finite support. We represent the combination of a binarized probability distribution and a probability level by a consistent pdBf which can then be compactly extended as an isotone Boolean function. The Boolean extension represents the sufficient requirements for a probabilistic constraint to hold, and is modelled as a DNF including p-patterns. Each p-pattern is a conjunction of literals and defines sufficient conditions for the satisfaction of a probabilistic constraint.

Enumerative methods, which are most often used for pattern generation purposes, are not very efficient for the construction of patterns with large degree. This motivates the design of a mathematical programming method for pattern generation. Four formulations (2 LPs and 2 IPs) are proposed for the

Table 6 Supply Chain Instances - Solution Times and Optimality Gap

Instance Type			IP1		IP2		LP1		LP2		DEIP
$\|J\|$	$\|\Omega\|$	p	Time	Opt. Gap	Time	Opt. Gap	Time	Opt. Gap	Time	Opt. Gap	Time
10	5000	0.875	1.594	1.316%	0.406	1.449%	0.030	1.698%	0.031	0.802%	2.212
10	5000	0.9	0.109	0.458%	0.063	0.253%	0.016	0.485%	0.016	0.000%	0.063
10	5000	0.925	0.031	0.734%	0.031	0.971%	0.034	0.856%	0.031	0.560%	0.016
10	5000	0.95	0.031	0.325%	0.031	0.346%	0.015	0.457%	0.016	0.116%	0.005
10	10000	0.875	1.625	0.986%	0.391	0.960%	0.020	1.442%	0.016	1.292%	0.141
10	10000	0.9	0.297	0.396%	0.125	0.432%	0.031	0.421%	0.031	0.868%	0.078
10	10000	0.925	0.031	0.778%	0.016	0.939%	0.036	0.967%	0.031	0.000%	0.016
10	10000	0.95	0.031	0.001%	0.016	0.001%	0.040	0.001%	0.031	0.001%	0.016
10	20000	0.875	1.594	0.292%	0.406	0.487%	0.015	0.541%	0.016	1.631%	0.109
10	20000	0.9	0.281	1.532%	0.094	1.509%	0.030	2.360%	0.031	1.341%	0.078
10	20000	0.925	0.031	0.352%	0.031	0.369%	0.014	0.796%	0.016	0.004%	0.031
10	20000	0.95	0.031	0.775%	0.031	1.031%	0.031	1.029%	0.031	0.002%	0.031
10	50000	0.875	1.578	0.231%	0.406	0.472%	0.016	0.237%	0.016	0.000%	0.141
10	50000	0.9	0.391	1.241%	0.141	1.525%	0.032	2.254%	0.031	1.033%	0.094
10	50000	0.925	0.031	0.381%	0.031	0.815%	0.015	1.259%	0.016	0.137%	0.016
10	50000	0.95	0.031	0.918%	0.031	0.846%	0.034	2.638%	0.031	0.853%	0.008
20	5000	0.875	14.563	0.796%	2.427	0.801%	0.221	1.234%	0.236	0.800%	1.036
20	5000	0.9	3.938	0.610%	1.313	0.608%	0.035	0.896%	0.031	0.459%	0.500
20	5000	0.925	0.047	0.260%	0.047	0.257%	0.036	0.364%	0.031	0.000%	0.032
20	5000	0.95	0.016	0.001%	0.031	0.000%	0.016	0.011%	0.016	0.126%	0.015
20	10000	0.875	15.698	0.924%	4.702	0.922%	0.256	1.264%	0.303	0.925%	1.874
20	10000	0.9	11.609	0.401%	3.688	0.406%	0.050	0.856%	0.031	0.001%	1.391
20	10000	0.925	0.047	0.205%	0.047	0.207%	0.020	0.745%	0.016	0.069%	0.031
20	10000	0.95	0.031	0.332%	0.016	0.340%	0.062	0.867%	0.031	0.068%	0.016
20	20000	0.875	26.539	0.904%	4.269	0.896%	0.126	1.762%	0.129	0.896%	2.891
20	20000	0.9	12.703	0.404%	2.281	0.408%	0.045	2.697%	0.047	0.878%	2.297
20	20000	0.925	0.016	0.175%	0.000	0.175%	0.015	1.671%	0.016	0.000%	0.031
20	20000	0.95	0.031	0.142%	0.016	0.137%	0.031	2.012%	0.031	0.137%	0.031
20	50000	0.875	28.963	0.862%	4.982	0.862%	1.784	3.002%	1.625	0.862%	2.652
20	50000	0.9	21.172	0.666%	3.578	0.678%	1.365	3.216%	1.438	0.956%	1.891
20	50000	0.925	0.047	0.459%	0.047	0.479%	0.010	2.891%	0.016	0.275%	0.031
20	50000	0.95	0.031	0.181%	0.047	0.189%	0.010	2.421%	0.016	0.252%	0.016

generation of p-patterns, which in turn allows for the derivation of a linear programming inner approximation of the probabilistic problem. Finally, we propose a model that allows for the simultaneous *(i)* derivation of the p-pattern defining the minimal conditions for the probabilistic constraint to hold and *(ii)* optimal solution of the deterministic equivalent problem.

The results for complex stochastic problems, in which a very fine discretization involving up to 50,000 scenarios is applied to represent the random variables, highlight the computational efficiency of the approach. All problems are solved to optimality in less than three seconds. Moreover, the solution time is not an increasing function of the number of realizations used to describe the random variables. Numerical results based on instances of two different problems (cash matching and supply chain) indicate that the linear programming formulation LP2 is solved the fastest and provides the tightest inner approximations (i.e., for more than 60% of the problem instances). The formulation LP2 is based on the pricing of the literals included in the p-pattern. This study offers the possibility to solve either a tight inner approximation or the deterministic equivalent of the probabilistic problem (1). The choice between both could depend on the specifics of the problem and of the decision-making process and stage.

The proposed approach can be applied in the exact same fashion in the cases when the stochastic problem (1) includes integer decision variables, contains non-linear constraint(s) and/or objective function, or when the inequality on which the probabilistic requirement is imposed is nonlinear. Extensions of the proposed approach could concern probabilistic constraints with a random technology matrix and two-stage stochastic programming problems.

Acknowledgment: This work is supported by the Army Research Office under Grant # W911NF-09-1-0497. The author is grateful to the valuable comments and suggestions from two anonymous reviewers, and the associate editors of *Operations Research* and the *Stochastic Programming E-Print Series* (*SPEPS*), who helped improve the paper significantly.

References

[1] Alexe G. 2007. *LAD - Datascope V2.1 Software Package*.

[2] Alexe G., Hammer P.L. 2006. Spanned Patterns for the Logical Analysis of Data. *Discrete Applied Mathematics* 154 (7), 1039-1049.

[3] Beraldi P., Ruszczyński A. 2002. A Branch and Bound Method for Stochastic Integer Problems under Probabilistic Constraints. *Optimization Methods and Software* 17 (3), 359-382.

[4] Birge J.R., Louveaux F. 1997. *Introduction to Stochastic Programming*. Springer, New York, NY.

[5] Boros E., Hammer P.L., Ibaraki T., Kogan A. 1997. Logical Analysis of Numerical Data. *Mathematical Programming* 79, 163-190.

[6] Boros E., Hammer P.L., Ibaraki T., Kogan A., Mayoraz E., Muchnik I. 2000. An Implementation of Logical Analysis of Data. *IEEE Transactions on Knowledge and Data Engineering* 12(2), 292-306.

[7] Calafiore G.C., Campi M.C. 2005. Uncertain Convex Programs: Randomized Solutions and Confidence Levels. *Mathematical Programming* 102, 25-46.

[8] Center for Research in Security Prices. CRSP US Treasury Database: `http://www.crsp.com/products/treasuries.htm`.

[9] Charnes A., Cooper W.W., Symonds G.H. 1958. Cost Horizons and Certainty Equivalents: An Approach to Stochastic Programming of Heating Oil. *Management Science* 4, 235-263.

[10] Cheon M.-S., Ahmed S., Al-Khayyal F. 2006. A Branch-Reduce-Cut Algorithm for the Global Optimization of Probabilistically Constrained Linear Programs. *Mathematical Programming* 108, 617-634.

[11] Crama Y., Hammer P.L. 2011. *Boolean Functions - Theory, Algorithms, and Applications*. Eds: Cambridge Press University, New York, NY.

[12] Dentcheva D., Lai B., Ruszczyński A. 2004. Dual Methods for Probabilistic Optimization Problems. *Mathematical Methods of Operations Research* 60 (2), 331-346.

[13] Dentcheva D., Martinez G. 2012. Augmented Lagrangian Method for Probabilistic Optimization. *Annals of Operations Research*. Forthcoming: `http://www.springerlink.com/content/n0j09h7156052r31/`.

[14] Dentcheva D., Prékopa A., Ruszczyński A. 2001. Concavity and Efficient Points of Discrete Distributions in Probabilistic Programming. *Mathematical Programming* 47 (3), 1997-2009.

[15] Djukova E., Inyakin A.S., Peskov N.V., Sakharov A.A. 2006. Increasing the Efficiency of Combinatorial Logical Data Analysis in Recognition and Classification Problems. *Pattern Recognition and Image Analysis* 16 (4), 707-711.

[16] Djukova E., Zhuravlev Y. 2000. Discrete Analysis of Feature Descriptions in Recognition Problems of High Dimensionality. *Computational Mathematics and Mathematical Physics* 40, 1214-1227.

[17] Duda R. O., Stork D.G., Hart P.E. 2001. *Pattern Classification*, Second Edition. Wiley & Sons.

[18] Eiter T., Ibaraki T., Makino K. 2002. Decision Lists and Related Boolean Functions. *Theoretical Computer Science* 270 (1-2), 493-524.

[19] Fourer R., Gay D.M., Kernighan B.W. 2003. *AMPL: A Modeling Language for Mathematical Programming*. Second Edition, Duxbury Press Brooks Cole Publishing Co.

[20] Fukunaga K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press. London, UK.

[21] Gren I.-M. 2008. Adaptation and Mitigation Strategies for Controlling Stochastic Water Pollution: An Application to the Baltic Sea. *Ecological Economics* 66 (2-3), 337-347.

[22] Grenander U., Miller M.I. 2007. *Pattern Theory: From Representation to Inference*. Oxford University Press, NY, USA.

[23] Hammer P.L. 1986. Partially Defined Boolean Functions and Cause-Effect Relationships. *International Conference on Multi-Attribute Decision Making Via OR-Based Expert Systems*. University of Passau, Passau, Germany.

[24] Hammer P.L., Bonates T.O. 2006. Logical Analysis of Data - An Overview: From Combinatorial Optimization to Medical Applications. *Annals of Operations Research* 148 (1), 203-225.

[25] Hammer P.L., Kogan A., Lejeune M.A. 2006. Modeling Country Risk Ratings Using Partial Orders. *European Journal of Operational Research* 175 (2), 836-859.

[26] Hammer P.L., Kogan A., Simeone B., Szedmak S. 2004. Pareto-Optimal Patterns in Logical Analysis of Data. *Discrete Applied Mathematics* 144, 79-102.

[27] Henrion R. 2004. Introduction to Chance-Constrained Programming. *Tutorial Paper for the Stochastic Programming Community Home Page*.

[28] Hooker J. 2000. *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*. Wiley & Sons.

[29] Hooker J.N. 2007. *Integrated Methods for Optimization*. Springer. New York, NY, USA.

[30] Ibaraki T. 2011. *Partially Defined Boolean Functions*. In: *Boolean Functions - Theory, Algorithms, and Applications.*. Eds: Crama Y., Hammer P.L. Cambridge Press University, 511-563.

[31] Jeroslow R.G. 1989. *Logic-Based Decision Support: Mixed Integer Model Formulation*. Annals of Discrete Mathematics. North-Holland. Amsterdam, The Netherlands.

[32] Kress M., Penn M., Polukarov M. 2007. The Minmax Multidimensional Knapsack Problem with Application to a Chance-Constrained Problem. *Naval Research Logistics* 54, 656-666.

[33] Küçükyavuz S. 2012. On Mixing Sets Arising in Probabilistic Programming. *Mathematical Programming* 132, 31-56.

[34] Lejeune M.A. 2008. Preprocessing Techniques and Column Generation Algorithms for p-Efficiency. *Journal of Operational Research Society* 59, 1239-1252.

[35] Lejeune M.A. 2012. Pattern Definition of the p-Efficiency Concept. *Annals of Operations Research*. Forthcoming: `http://www.springerlink.com/content/g067806p14846854/`.

[36] Lejeune M.A., Noyan N. 2010. Mathematical Programming Approaches for Generating p-Efficient Points. *European Journal of Operational Research* 207 (2), 590-600.

[37] Lejeune M.A., Ruszczyński A. 2007. An Efficient Trajectory Method for Probabilistic Inventory-Production-Distribution Problems. *Operations Research* 55 (2), 378-394.

[38] Luedtke J., Ahmed S. 2008. A Sample Approximation Approach for Optimization with Probabilistic Constraints. *SIAM Journal of Optimization* 19, 674-699.

[39] Luedtke J., Ahmed S., Nemhauser G. 2010. An Integer Programming Approach for Linear Programs with Probabilistic Constraints. *Mathematical Programming* 122 (2), 247-272.

[40] Martinez-Trinidad J. F., Guzmán-Arenas A. 2001. The Logical Combinatorial Approach to Pattern Recognition, an Overview Through Selected Works. *Pattern Recognition* 34 (4), 741-751.

[41] Mergent Inc. Mergent Fixed Income Securities Database: `http://www.mergent.com/findSolution-fixedIncome.html`.

[42] Miller B.L., Wagner H.M. 1965. Chance Constrained Programming with Joint Constraints. *Operations Research* 13, 930-945.

[43] Nemirovski A., Shapiro A. 2006. Convex Approximations of Chance Constrained Programs. *SIAM Journal on Optimization* 17, 969-996.

[44] Prékopa A. 1970. On Probabilistic Constrained Programming. *Proceedings of the Princeton Symposium on Mathematical Programming*. Princeton University Press, 113-138.

[45] Prékopa A. 1973. Contributions to the Theory of Stochastic Programming. *Mathematical Programming* 4, 202-221.

[46] Prékopa A. 1990. Dual Method for a One-Stage Stochastic Programming with Random rhs Obeying a Discrete Probability Distribution. *Zeitschrift of Operations Research* 34, 441-461.

[47] Prékopa A. 1995. *Stochastic Programming*. Kluwer. Boston, MA.

[48] Prékopa A. 2003. Probabilistic Programming Models. Chapter 5 in: *Stochastic Programming: Handbook in Operations Research and Management Science 10*. Eds: Ruszczyński A., Shapiro A. Elsevier Science Ltd, 267-351.

[49] Prékopa A., Vizvari B., Badics T. 1998. Programming under Probabilistic Constraint with Discrete Random Variable. In: *New Trends in Mathematical Programming*. Eds: Giannessi F., Komlósi S., Rapcsák T. Boston, MA.

[50] Radeanu S. 1974. *Boolean Functions and Equations*. North-Holland, Amsterdam.

[51] Ruiz-Shulcloper J., Abidi M.A. 2002. Logical Combinatorial Pattern Recognition: A Review. *Transworld Research Networks* 3, 133-176.

[52] Ruszczyński A. 2002. Probabilistic Programming with Discrete Distribution and Precedence Constrained Knapsack Polyhedra. *Mathematical Programming* 93, 195-215.

[53] Ruszczyński A., Shapiro A. 2003. *Stochastic Programming: Handbook in Operations Research and Management Science* 10. Elsevier Science Ltd.

[54] Saxena A., Goyal V., Lejeune M.A. 2010. MIP Reformulations of the Probabilistic Set Covering Problem. *Mathematical Programming* 121 (1), 1-31.

[55] Sen S. 1992. Relaxations for Probabilistically Constrained Programs with Discrete Random Variables. *Operations Research Letters* 11, 81-86.

[56] Tanner M.W., Ntaimo L. 2010. IIS Branch-and-Cut for Joint Chance-Constrained Stochastic Programs and Application to Optimal Vaccine Allocation. *European Journal of Operational Research* 207, 290-296.

[57] Torvik V.I., Triantaphyllou E. 2009. Inference of Monotone Boolean Functions. *Encyclopedia of Optimization*, 1591-1598.

[58] Triantaphyllou E., Felici G. 2006. *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques.* Springer, Heidelberg, Germany.

[59] Truemper K. 2004. *Design of Logic-Based Intelligent Systems*. Wiley & Sons, Hoboken, NJ.

[60] Urbano R.H., Mueller R.K. 1956. A Topological Method for the Determination of the Minimal Forms of a Boolean Function. *IRE Transactions on Electronic Computers* EC-5, 126-132.

[61] Vapnik V.N. 1998. *Statistical Learning Theory*. Wiley & Sons, USA.

Table 7 Cash Matching Instances - Solution Times and Optimality Gap

Instance Type			IP1		IP2		LP1		LP2		DEIP
$\lvert J \rvert$	$\lvert\Omega\rvert$	p	Time	Opt. Gap	Time	Opt. Gap	Time	Opt. Gap	Time	Opt. Gap	Time
10	5000	0.875	1.231	1.401%	0.324	1.210%	0.024	1.745%	0.025	0.873%	1.064
10	5000	0.9	0.125	0.521%	0.031	0.321%	0.010	0.667%	0.010	0.204%	0.038
10	5000	0.925	0.102	0.602%	0.052	0.844%	0.031	0.962%	0.031	0.521%	0.024
10	5000	0.95	0.031	0.147%	0.016	0.152%	0.011	0.322%	0.012	0.163%	0.031
10	10000	0.875	1.128	1.150%	0.287	0.816%	0.024	1.309%	0.016	0.896%	0.084
10	10000	0.9	0.241	0.431%	0.057	0.431%	0.033	0.458%	0.024	0.434%	0.056
10	10000	0.925	0.062	0.290%	0.031	0.257%	0.026	0.602%	0.028	0.235%	0.021
10	10000	0.95	0.038	0.201%	0.016	0.327%	0.030	0.308%	0.033	0.246%	0.038
10	20000	0.875	1.280	0.229%	0.312	0.356%	0.042	0.559%	0.063	0.278%	0.070
10	20000	0.9	0.302	0.942%	0.052	1.007%	0.031	1.620%	0.025	0.708%	0.042
10	20000	0.925	0.052	0.224%	0.014	0.209%	0.012	0.429%	0.010	0.209%	0.011
10	20000	0.95	0.049	0.791%	0.016	0.652%	0.031	1.098%	0.035	0.324%	0.058
10	50000	0.875	1.432	0.234%	0.239	0.117%	0.114	0.311%	0.135	0.136%	0.078
10	50000	0.9	0.202	0.968%	0.049	1.122%	0.028	1.964%	0.034	0.739%	0.081
10	50000	0.925	0.059	0.159%	0.031	0.264%	0.012	0.238%	0.012	0.128%	0.010
10	50000	0.95	0.016	0.936%	0.016	0.735%	0.008	1.427%	0.010	0.748%	0.035
15	5000	0.875	8.645	0.484%	0.987	0.452%	0.156	0.487%	0.214	0.463%	0.421
15	5000	0.9	2.004	0.725%	0.857	0.712%	0.154	1.023%	0.231	0.427%	0.095
15	5000	0.925	0.105	0.018%	0.042	0.081%	0.020	0.124%	0.024	0.007%	0.032
15	5000	0.95	0.028	0.015%	0.030	0.014%	0.015	0.016%	0.012	0.015%	0.015
15	10000	0.875	9.362	0.745%	3.214	0.637%	0.361	1.326%	0.340	0.641%	0.757
15	10000	0.9	7.203	0.527%	3.110	0.520%	0.284	0.943%	0.352	0.254%	1.125
15	10000	0.925	0.098	0.089%	0.068	0.163%	0.018	0.283%	0.015	0.097%	0.039
15	10000	0.95	0.053	0.173%	0.038	0.145%	0.012	0.715%	0.016	0.202%	0.025
15	20000	0.875	17.365	0.854%	3.860	0.748%	0.421	1.527%	0.257	0.716%	2.125
15	20000	0.9	8.322	0.257%	1.236	0.336%	0.120	0.637%	0.163	0.316%	1.741
15	20000	0.925	0.134	0.129%	0.016	0.102%	0.010	0.384%	0.012	0.074%	0.204
15	20000	0.95	0.102	0.184%	0.016	0.224%	0.018	1.427%	0.015	0.004%	0.087
15	50000	0.875	21.312	0.579%	4.236	0.617%	1.021	2.127%	1.124	0.472%	2.035
15	50000	0.9	14.329	0.719%	2.810	0.641%	0.752	3.360%	0.702	0.598%	0.859
15	50000	0.925	0.521	0.208%	0.441	0.234%	0.102	1.986%	0.084	0.225%	0.358
15	50000	0.95	0.385	0.064%	0.321	0.082%	0.077	1.057%	0.065	0.028%	0.303