Math. Program., Ser. A
DOI 10.1007/s10107-008-0224-y

FULL LENGTH PAPER

MIP reformulations of the probabilistic set covering problem

**Anureet Saxena · Vineet Goyal ·
Miguel A. Lejeune**

Received: 6 February 2007 / Accepted: 18 March 2008

Abstract In this paper, we address the following probabilistic version (PSC) of the set covering problem: $\min\{cx \mid \mathbb{P}(Ax \geq \xi) \geq p,\ x \in \{0, 1\}^N\}$ where A is a 0-1 matrix, ξ is a random 0-1 vector and $p \in (0, 1]$ is the threshold probability level. We introduce the concepts of *p-inefficiency* and polarity cuts. While the former is aimed at deriving an equivalent MIP reformulation of (PSC), the latter is used as a strengthening device to obtain a stronger formulation. Simplifications of the MIP model which result when one of the following conditions hold are briefly discussed: A is a balanced matrix, A has the circular ones property, the components of ξ are pairwise independent, the distribution function of ξ is a *stationary* distribution or has the *disjunctive shattering* property. We corroborate our theoretical findings by an extensive computational experiment on a test-bed consisting of almost 10,000 probabilistic instances. This test-bed was created using deterministic instances from the literature and consists of probabilistic variants of the set covering model and capacitated versions of facility location, warehouse location and k-median models. Our computational results show that our procedure is orders of magnitude faster than any of the existing approaches to solve (PSC), and in many cases can reduce hours of computing time to a fraction of a second.

Anureet Saxena's research was supported by the National Science Foundation through grant #DMI-0352885 and by the Office of Naval Research through contract N00014-03-1-0133.
Vineet Goyal's research was supported in part by NSF grant CCF-0430751 and ITR grant CCR-0122581.

A. Saxena · V. Goyal
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: anureets@andrew.cmu.edu

V. Goyal
e-mail: vgoyal@andrew.cmu.edu

M. A. Lejeune (✉)
LeBow College of Business, Drexel University, Philadelphia, PA 19104, USA
e-mail: mlejeune@drexel.edu

Keywords Probabilistic programming · Set covering · Mixed integer programming · Cutting planes

Mathematics Subject Classification (2000) 90C15 · 90-08 · 90C10

1 Introduction

In this paper, we address the following probabilistic variant of the set covering problem,

$$\begin{aligned} \min \quad & cx \\ s.t \quad & \\ & \mathbb{P}(Ax \geq \xi) \geq p \\ & x_j \in \{0, 1\} \; j \in N \end{aligned} \tag{PSC}$$

where A is a 0-1 matrix defined on row-index set M and column-index set N, ξ is a 0-1 random M-vector, $p \in (0, 1]$ is the value of the *threshold probability* (also called the *reliability level*) and $c \in \mathbb{R}^N$ is the cost vector. Indeed, if we replace the probabilistic constraint $\mathbb{P}(Ax \geq \xi) \geq p$ in (PSC) by $Ax \geq 1$ we recover the well-known set covering problem.

(PSC) belongs to a class of optimization problems commonly referred to as probabilistic programs. Probabilistic programming was introduced by Charnes and Cooper [7] in the late fifties and has since then been studied extensively [19]. We refer the reader to Prékopa [20] for a review of recent developments in this area. (PSC) is a very challenging problem in the field of stochastic mixed integer programming which combines inherent complexity of both mixed integer programming and stochastic programming. Several set covering models which can be solved in a matter of seconds by state-of-art MIP solvers (such as CPLEX or XPRESS) can give rise to probabilistic problems which can take several minutes (at times hours) to solve [5]. One of the notions which has played a pivotal role in the algorithmic development of (PSC) is that of *p-efficiency*. Originally introduced by Prékopa [18], the concept of p-efficiency of a discrete probability distribution has been the focus of intense research in recent years [5,6,12,16,21].

In a recent development, Beraldi and Ruszczyński [5] proposed an algorithm to solve (PSC). Their algorithm involves enumerating the complete set of p-efficient points of the distribution, and then solving a deterministic set covering problem for each one of the p-efficient points. Some discrete distributions can have an extremely large number of p-efficient points, even at a high reliability level, which makes the enumeration phase very expensive. Solving MIPs from each one of these p-efficient points is a different proposition altogether. Beraldi and Ruszczyński [5] experimented with some hybrid techniques to improve their algorithm, but concluded that the enumeration of p-efficient points continued to be the bottleneck in their procedure.

In this paper, we focus on the case when the random variable ξ can be decomposed into L blocks say $\{\xi^1, \ldots, \xi^L\}$ such that ξ^t is a 0-1 random M_t-vector for $t \in \{1, \ldots, L\}$ (where $M_1, \ldots, M_L$ is a partition of M), and ξ^i and ξ^j are independent random vectors for distinct i, j. While the idea of using random variables which have this kind of block structure was originally proposed by Beraldi and Ruszczyński [5] to construct a test-bed of problem instances for (PSC), its detailed theoretical and computational investigation is presented for the first time in the current paper. In particular, we show that in the presence of this additional structure the enumeration phase in the algorithm of [5] can be encoded as a mixed integer program thereby combining (*MIPing*[1]) the two phases into one integrated MIP. Expressing the enumeration problem as a mixed integer program allows us to use a state-of-art MIP solver to perform efficient enumeration thereby reaping the benefits of developments in the field of mixed-integer programming. Indeed, our computational experiments conducted over a test-bed of almost 10,000 probabilistic instances demonstrate that (PSC) derived from simple and moderately difficult set covering problems can themselves be formulated as simple or moderately difficult MIPs. As a byproduct of our research, we introduce the concept of p-inefficiency and polarity cuts. While the former is aimed at reducing the number of constraints in our model, the latter is used as a strengthening device to obtain stronger formulations.

Recently, Luedtke et al. [17] studied the linear programming problem with probabilistic right-hand sides. While being closely related, there are two crucial differences between our work and the one presented in [17]. First, the underlying deterministic problem in our case, namely the set covering problem, is itself a NP-hard problem, whereas the corresponding problem in the framework of [17] is the polynomially solvable linear programming problem. Second, we do not make any assumptions about the distribution function of ξ^t except that the underlying random variable is 0-1 valued; Luedtke et al. [17], on the other hand, place no restrictions on the domain of the random variable but assume that it has a finite distribution.

The rest of the paper is organized as follows. In Sect. 2 we introduce the notion of p-inefficiency and use it to construct a MIP reformulation of (PSC). We also discuss simplifications of the MIP reformulation which arise when A is a balanced matrix or has the circular ones property. In Sect. 3 we discuss a class of cutting planes for (PSC), which we refer to as polarity cuts. We derive a linear programming based separation algorithm for these cuts and discuss techniques to reduce the coefficient matrix densities of the resulting linear programs. The MIP reformulation introduced in Sect. 2 involves enumeration of the p-inefficient frontier of the distribution of ξ_t for $t \in \{1 \dots L\}$. Special properties of the distribution functions can at times be used to circumvent this enumeration phase. In Sect. 4 we discuss one such distribution, namely the stationary distribution, and give a compact (polynomial sized) MIP reformulation of (PSC) for this special case. We generalize the notion of stationary distribution and characterize a large class of distributions which share this property. Finally, we discuss our computational results in Sect. 5. We corroborate our theoretical findings by a computational experiment conducted on a test-bed consisting of almost 10,000 probabilistic instances. This test-bed was constructed from probabilistic variants of set covering models and capacitated versions of facility location, warehouse location and k-median models. In Sect. 6 we present some concluding remarks. Detailed computational results and proofs of some of the results presented in this paper can be found in [22].

---

[1] The phrase *MIPing* was coined by Matteo Fischetti and Andrea Lodi at the Ninth International meeting on Combinatorial Optimization (2005) at Aussois, France.

2 MIP formulation

In this section we discuss a mixed integer programming formulation of (PSC). Henceforth, for $z \in \mathbb{R}^M$ we denote by z^t the sub-vector of z formed by components in M_t for $t = 1 \dots L$. Furthermore, let $F : \{0, 1\}^M \to \mathbb{R}$ denote the cumulative distribution function of ξ and let F_t denote the restriction of F to M_t for $t = 1 \dots L$. In other words, for $z \in \{0, 1\}^M$, $F(z) = \mathbb{P}(\xi \leq z)$ and $F_t(z^t) = \mathbb{P}(\xi^t \leq z^t)$. In order to express the formulation succinctly we need the following definition.

Definition 2.1 A point $v \in \{0, 1\}^m$ is called a *p-inefficient* point of the probability distribution function F if $F(v) < p$ and there is no binary point $w \geq v,\ w \neq v$ such that $F(w) < p$. The set of all p-inefficient points of F is called the *p-inefficient frontier* of F.

The notion of p-inefficiency is closely related to the notion of p-efficiency introduced by Prékopa [18]. A point $v \in \{0, 1\}^m$ is called a *p-efficient* point of the discrete probability distribution function F if $F(v) \geq p$ and there is no binary point $w \leq v,\ w \neq v$ such that $F(w) \geq p$. The set of all p-efficient points of F is called the *p-efficient frontier* of F. If S is the set of binary vectors which are either p-efficient or dominate[2] a p-efficient point, and T is the set of binary vectors which are either p-inefficient or are dominated by a p-inefficient point, then $\{S, T\}$ defines a partition of the lattice $\{0, 1\}^m$.

For $t \in \{1 \dots L\}$, let S_t denote the set of binary vectors which are either p-efficient or dominate a p-efficient point of F_t and let I_t denotes the set of p-inefficient points of F_t. The theorem that follows gives a MIP reformulation of (PSC).

Theorem 2.2 (PSC) *can be reformulated as the following mixed integer program.*

$$
\begin{array}{llr}
\min_{(x,z,\eta)} & cx & \\
s.t & & \\
& Ax \geq z & \\
& \sum_{t=1}^{L} \eta_t \geq ln\ p & \\
& \eta_t \leq (ln\ F_t(v)) \left(1 - \sum_{i \in M_t, v_i = 0} z_i\right) \quad \forall v \in S_t\ \forall t \in \{1 \dots L\} & \text{(MIP1)} \\
& 1 \leq \sum_{i \in M_t, v_i = 0} z_i \quad \forall v \in I_t\ \forall t \in \{1 \dots L\} & \\
& x_j \in \{0, 1\}\ \forall j \in N & \\
& z_i \in \{0, 1\}\ \forall i \in M &
\end{array}
$$

Proof Suppose x is a feasible solution to (PSC). Let $z_i = \min(a_i^T x, 1)\ i \in M$, where a_i denotes the ith row of A; $z_i = 1$ if and only if the ith row of A is covered by x. Let $\eta_t = ln\ F_t(z^t)$ for $t \in \{1 \dots L\}$. Note that $p \leq \mathbb{P}(Ax \geq \xi) = \mathbb{P}(z \geq \xi) = \Pi_{t=1}^{L} F_t(z^t)$ and $ln\ p \leq \sum_{t=1}^{L} ln\ F_t(z^t) = \sum_{t=1}^{L} \eta_t$. Let $t \in \{1 \dots L\}$ and $v \in S_t \cup I_t$. If $v \in I_t$ then $z^t \not\leq v$ (since $F_t(z^t) \geq p$) and $\sum_{i \in M_t, v_i = 0} z_i \geq 1$. If $v \in S_t$ and $\sum_{i \in M_t, v_i = 0} z_i \geq 1$ then $\eta_t \leq (ln\ F_t(v)) \left(1 - \sum_{i \in M_t, v_i = 0} z_i\right)$ is trivially satisfied; if $\sum_{i \in M_t, v_i = 0} z_i = 0$ then $z^t \leq v$ and $\eta_t = ln\ F_t(z^t) \leq ln\ F_t(v)$. Hence (x, z, η) is a feasible solution to (MIP1).

[2] If $x, y \in \{0, 1\}^M$ then x is said to *dominate* y if $x \geq y$.

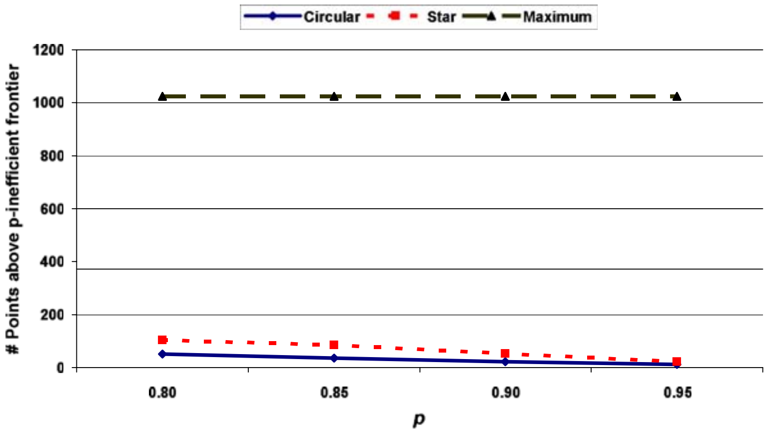

Fig. 1 Number of points above the p-inefficient frontier

Conversely, suppose (x, z, η) is a feasible solution to (MIP1). If $z^t \notin S_t$ then $\exists v \in I_t$ such that $z^t \leq v$, $\sum_{i \in M_t, v_i = 0} z_i = 0$, contradicting the feasibility of (x, z, η); hence $z^t \in S_t$ $\forall t \in \{1 \dots L\}$. By choosing $v = z^t \in S_t$, we get $\eta_t \leq (ln\ F_t(v))\big(1 - \sum_{i \in M_t, v_i = 0} z_i\big) = ln\ F_t(z^t)$. Since $ln\ p \leq \sum_{t=1}^{L} \eta_t \leq \sum_{t=1}^{L} ln\ F_t(z^t)$, it follows that $p \leq \Pi_{t=1}^{L} F_t(z^t) = F(z) = \mathbb{P}(Ax \geq \xi)$, and x is a feasible solution to (PSC). □

Note that (MIP1) has constraints arising only from points which are either p-inefficient or dominate a p-inefficient point of F_t. In theory, the number of such points can be exponential; however, in practice the number of such points is only a small fraction of the total number of lattice points.

Figure 1 shows this information graphically for two distributions, namely Circular and Star (see [5] for the definition of these distributions). The horizontal axis represents the reliability level p, whereas the vertical axis gives the number of points which are either p-inefficient or dominate a p-inefficient point, averaged over 1,000 randomly generated instantiations (block size = 10) of the Circular and Star distributions. Note that, even at a reliability level of 0.8 less than 20% of the lattice points qualified the above condition. As an interesting consequence of this observation, it follows that our model can handle significantly large block sizes provided there exists an efficient algorithm to enumerate points which lie on or above the the p-inefficient frontier. Such an enumeration algorithm, clearly, would need to exploit properties of the specific distribution involved and its investigation goes beyond the scope of this paper.

Let (MIP1') denote the relaxation of (MIP1) obtained by replacing the integrality conditions on z variables by $0 \leq z \leq 1$. Let opt(MIP1) and opt(MIP1') denote the optimal values of (MIP1) and (MIP1'), respectively. The proposition that follows shows that (MIP1') is also a valid reformulation of (PSC).

Proposition 2.3 $opt(MIP1) = opt(MIP1')$.

Proof It suffices to show that $opt(MIP1) \leq opt(MIP1')$. Suppose (x, z, η) is an optimal solution to (MIP1'). Let $\bar{z} \in \{0, 1\}^M$ be defined as, $\bar{z}_i = 1$ if and only if

$a_i^T x \geq 1$ where a_i denotes the i^{th} row of A for $i \in M$. Note that for $v \in S_t \cup I_t$ ($t = 1 \ldots L$), $\sum_{i \in M_t, v_i = 0} z_i \leq \sum_{i \in M_t, v_i = 0} \bar{z}_i$, which implies that $(x, \bar{z}, \eta)$ is also a feasible solution to (MIP1), and $opt(MIP1) \leq opt(MIP1')$. □

(MIP1) and (MIP1') provide two contrasting alternatives to solve (PSC). While (MIP1') has fewer number of binary variables, the integrality constraints on the z variables in (MIP1) can be used to generate strong valid cutting planes which might assist the overall solution procedure. We used a hybrid model in our experiments which combines the attractive features of both of these models. In Sect. 3 we discuss a class of strong valid cutting planes for the (MIP1) formulation, called polarity cuts, which do not cut off the optimal solution to (MIP1'). In our computational experiments we strengthened the (MIP1') formulation at the root node by these polarity cuts, and then applied CPLEX to the strengthened formulation. As our computational results (Sect. 5) demonstrate, this hybrid model is substantially better than (MIP1) or (MIP1'), considered individually.

There is a crucial difference between our approach and the one pursued in [5]. For the sake of discussion, consider the case of one block ($L = 1$). In this case, while both of these approaches entail enumeration of the p-efficient frontier [5] or of the points which lie on or above the p-inefficient frontier (MIP1'), the manner in which the enumerated points are used in the respective algorithm is entirely different. The solution approach in [5] potentially involves the solution of a mixed-integer program for every point on the p-efficient frontier whereas our approach favors utilizing these points in constructing an equivalent MIP reformulation (MIP1') of (PSC). Alternatively, (MIP1') can be viewed as encoding the enumerative aspect in the algorithm of [5] by using auxiliary *continuous* variables (z, η) thereby facilitating the MIP solver to perform efficient enumeration. Indeed as the number of blocks L increases, the approach in [5] becomes prohibitively expensive as compared to our approach, since it involves computation of the p-efficient frontier of the distribution of ξ which can grow exponentially even if $|S_t \cup I_t|$ is polynomially bounded for $t \in \{1 \ldots L\}$ (see Sect. 5).

Next we illustrate Theorem 2.2 on a small example. Consider the following probabilistic set covering problem in which the reliability level p is equal to 0.8 and the right-hand side is a 5-dimensional random 0-1 vector whose cumulative distribution function is given in Table 1.

$$\begin{aligned} &\min \sum_{j=1}^{5} x_j \\ &s.t \\ &\mathcal{P}\begin{pmatrix} x_1 + x_2 \geq \xi_1 \\ x_2 + x_3 \geq \xi_2 \\ x_3 + x_4 \geq \xi_3 \\ x_4 + x_5 \geq \xi_4 \\ x_5 + x_1 \geq \xi_5 \end{pmatrix} \geq 0.8 \\ &x_j \in \{0, 1\}\ j = 1, \ldots, 5 \end{aligned} \tag{2.1}$$

Table 1 Cumulative probability distribution

0-1 Vector v	$\mathbb{P}(\xi \leq v)$	$ln\ \mathbb{P}(\xi \leq v)$	Type of v
00000	0.36834	−0.99876	T
00001	0.36834	−0.99876	T
00010	0.36834	−0.99876	T
00011	0.37243	−0.98769	T
00100	0.36834	−0.99876	T
00101	0.36834	−0.99876	T
00110	0.95672	−0.04424	S
00111	0.96736	−0.03318	S
01000	0.36834	−0.99876	T
01001	0.36834	−0.99876	T
01010	0.36834	−0.99876	T
01011	0.37243	−0.98769	T
01100	0.37243	−0.98769	T
01101	0.37243	−0.98769	T
01110	0.96736	−0.03318	S
01111	0.97812	−0.02212	S
10000	0.36834	−0.99876	T
10001	0.37243	−0.98769	T
10010	0.36834	−0.99876	T
10011	0.37658	−0.97663	T
10100	0.36834	−0.99876	T
10101	0.37243	−0.98769	T
10110	0.95672	−0.04424	S
10111	0.97812	−0.02212	S
11000	0.37243	−0.98769	T
11001	0.37658	−0.97663	T
11010	0.37243	−0.98769	T
11011	0.38077	−0.96557	I
11100	0.37658	−0.97663	T
11101	0.38077	−0.96557	I
11110	0.97812	−0.02212	S
11111	1.00000	0.00000	S

The first column of the table contains a 5-dimensional binary vector v, the second column gives the value of $\mathbb{P}(\xi \leq v)$ while the third column contains $ln\ \mathbb{P}(\xi \leq v)$. The fourth column categorizes the binary vector v into one of the following three categories; (S): v is either p-efficient or dominate a p-efficient point, (I): v is p-inefficient and (T): v is dominated by a p-inefficient point. The distribution represented by Table 1 has 8 points of type S and 2 points of type I. The (MIP1) formulation for this probabilistic instance is given by,

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{5} x_j \\
s.t \\
& x_1 + x_2 \geq z_1 \\
& x_2 + x_3 \geq z_2 \\
& x_3 + x_4 \geq z_3 \\
& x_4 + x_5 \geq z_4 \\
& x_5 + x_1 \geq z_5 \\
& \quad \eta_1 \geq -0.22314 \qquad\qquad (ln\ 0.8 = -0.22314) \\
& \quad \eta_1 \leq -0.04424(1 - z_1 - z_2 - z_5) \\
& \quad \eta_1 \leq -0.03318(1 - z_1 - z_2) \\
& \quad \eta_1 \leq -0.03318(1 - z_1 - z_5) \\
& \quad \eta_1 \leq -0.04424(1 - z_2 - z_5) \\
& \quad \eta_1 \leq -0.02212(1 - z_1) \\
& \quad \eta_1 \leq -0.02212(1 - z_2) \\
& \quad \eta_1 \leq -0.02212(1 - z_5) \\
& \quad \eta_1 \leq 0 \\
& \quad z_3 \geq 1 \\
& \quad z_4 \geq 1 \\
& \quad x_j \in \{0, 1\}\ j = 1 \ldots 5 \\
& \quad z_i \in \{0, 1\}\ i = 1 \ldots 5
\end{aligned}
\tag{2.2}
$$

In some special cases, structural properties of the matrix A can be used to design MIP reformulations of (PSC) with fewer number of binary variables as compared to (MIP1). The proposition that follows gives example of one such property. Recall that a 0-1 matrix A has the circular ones property if in every row either the ones or zeros are consecutive. Set covering models involving 0-1 matrices with circular ones property often arise in scheduling problems [3].

Proposition 2.4 *If A has the circular ones property, then* (*MIP1*) *can be reformulated as,* $(n = |N|)$

$$
\begin{aligned}
\min_{(x,z,\eta,y)} \quad & cx \\
s.t \\
Ax &\geq z \\
\textstyle\sum_{t=1}^{L} \eta_t &\geq ln\ p \\
\eta_t &\leq (ln\ F_t(v)) \left(1 - \textstyle\sum_{i \in M_t, v_i = 0} z_i\right) \quad \forall v \in S_t\ \forall t \in \{1 \ldots L\} \\
1 &\leq \textstyle\sum_{i \in M_t, v_i = 0} z_i \qquad\qquad\qquad\quad \forall v \in I_t\ \forall t \in \{1 \ldots L\} \\
\textstyle\sum_{j \in N} x_j &= y \\
x_j &\leq 1\ j \in N \\
x_j &\geq 0\ j \in N \\
z_i &\in \{0, 1\}\ \forall i \in M \\
y &\in \{0, 1, \ldots, n\}
\end{aligned}
\tag{2.3}
$$

A fundamental question in polyhedral combinatorics is to determine necessary and sufficient conditions under which optimal value of the relaxation $\mathcal{R}$ of a combinatorial optimization problem $\mathcal{P}$ coincides with the optimal value of $\mathcal{P}$. For instance, it is well known that the LP relaxation of a pure integer program has integer optimal solution for every choice of the cost vector and right-hand side, if and only if the coefficient matrix of the integer program is Totally Unimodular. The proposition that follows gives a similar result about (PSC). Recall that a 0-1 matrix A is *ideal* if the polytope $\{x \mid Ax \geq 1,\ 0 \leq x \leq 1\}$ is integral. Similarly, a 0-1 matrix A is *balanced* if every submatrix of A is ideal (Theorem 6.1 of Cornuejols [11]; see also [10]).

Proposition 2.5 *Let* (*R1*) *denote the relaxation of* (*MIP1*) *obtained by replacing the integrality constraints on x variables by $0 \leq x \leq 1$. Given a 0-1 matrix A, the following two statements are equivalent.*

1. *opt*(*R1*) = *opt*(*MIP1*) *for every cost vector c, cumulative distribution function F and threshold probability p.*
2. *A is a balanced matrix.*

3 Polarity cuts

For the sake of brevity, we assume in this section that (PSC) has only one block ($L = 1$), unless otherwise stated. The results discussed here can be easily extended to the more general case ($L \geq 1$) by applying them to each one of the blocks independently. Recall that I denotes the set of p-inefficient points of F and S denotes the set of 0-1 points which are either p-efficient or dominate a p-efficient point of F. Consider the following set of constraints which constitute (MIP1).

$$\eta \leq (ln\ F(v)) \left(1 - \sum_{i \in M, v_i = 0} z_i\right)\ \forall v \in S \tag{3.4}$$

$$\sum_{i \in M, v_i = 0} z_i \geq 1\ \forall v \in I \tag{3.5}$$

$$z_i \in \{0, 1\}\ \forall i \in M \tag{3.6}$$

Let $P = \text{clconv}\{(z, \eta) \mid z \in \{0, 1\}^M,\ \eta \leq ln\ F(z),\ F(z) \geq p\}$ denote the closed convex hull of the set of points (z, η) which satisfy (3.4–3.6). A central question in polyhedral analysis is to examine the strength of the defining inequalities (3.4), (3.5) with respect to the underlying integer hull P.

For the sake of discussion, consider the example (2.1) introduced in Sect. 2. We generated the following complete minimal description of P using the PORTA [8] software.

$$\eta \leq -0.04424 + 0.02212z_1 + 0.01106z_2 + 0.01106z_5$$
$$\eta \leq -0.04424 + 0.01106z_1 + 0.02212z_2 + 0.01106z_5$$
$$\eta \leq -0.04424 + 0.01106z_1 + 0.01106z_2 + 0.02212z_5$$

$$
\begin{aligned}
\eta &\leq -0.04424 + 0.02212z_2 + 0.02212z_5 \\
z_3 &= 1 \\
z_4 &= 1 \\
0 \leq z_1 &\leq 1 \\
0 \leq z_2 &\leq 1 \\
0 \leq z_5 &\leq 1
\end{aligned}
$$

Each one of the above inequalities defines a facet of P. Note that the constraints in the MIP1 formulation (2.2) derived from points in S do not define facets of P; in fact, they do not even define non-empty faces of P. Indeed the constraint $\eta_1 \leq -0.04424(1 - z_1 - z_2 - z_5)$ (derived from $(0, 0, 1, 1, 0, -0.04424) \in P$) is strictly dominated by the facet-defining inequality $\eta \leq -0.04424 + 0.02212z_1 + 0.01106z_2 + 0.01106z_5$. This suggests that the inequalities derived from points in S can be significantly strengthened by coefficient tightening procedures. Strengthening inequalities by coefficient tightening has two shortcomings. First, such procedures are sequence dependent and produce different inequalities depending on the order in which the coefficients are examined. Consequently, several different inequalities can be obtained by strengthening a single inequality, and it is difficult to decide a priori which of these will strengthen the formulation most effectively. Second, such procedures can generate only a subset of all valid (facet-defining) inequalities of P. Next we describe a procedure to generate valid inequalities of P which overcomes both of these shortcomings.

The lemma that follows provides crucial insights into the polyhedral structure of P. Let $J = \{i \in M \mid z_i = 1\ \forall z \in \{0, 1\}^M\ s.t\ F(z) \geq p\}$. Let $e \in \{1\}^M$ denote a vector of ones and e^i $(i \in M)$ denote the ith unit vector.

Lemma 3.1 *For $i \in J$, $e - e^i$ is a p-inefficient point of F. Furthermore, $dim(P) = m + 1 - |J|$ and for $i \in M \setminus J$, $z_i \leq 1$ defines a facet of P. If $\alpha z - \beta\eta \geq \Delta$ defines a facet of P different from the ones defined by $z_i \leq 1$ $(i \in M \setminus J)$, then $\beta \geq 0$, $\alpha_i \geq 0\ \forall i \in M \setminus J$ and $\sum_{i \in M \setminus J} \alpha_i + \beta > 0$.*

Proof Clearly for $i \in J$, $F(e - e^i) < p$ and $e - e^i$ is a p-inefficient point of P. Furthermore, since $P \subseteq \{(z, \eta) \mid z_i = 1\ \forall i \in J\}$, $dim(P) \leq m + 1 - |J|$. To see that $dim(P) = m + 1 - |J|$, consider the following $m + 2 - |J|$ affinely independent points in P, $\{(e - e^i, ln\ F(e - e^i)) \mid i \in M \setminus J\} \cup \{(e, 0), (e, -1)\}$. Using a similar construction, it can be shown that $z_i \leq 1$ defines a facet of P for $i \in M \setminus J$. Suppose $\alpha z - \beta\eta \geq \Delta$ defines a facet of P different from the ones defined by $z_i \leq 1$ $(i \in M \setminus J)$. Since P recedes in the direction $(z = 0, \eta = -1)$, $\beta \geq 0$. For $i \in M \setminus J$, there exists $z \in \{0, 1\}^M$ and $\eta \in \mathbb{R}$ such that $(z, \eta) \in P$, $\alpha z - \beta\eta = \Delta$ and $z_i = 0$; since $(z + e^i, \eta) \in P$, $\alpha z + \alpha_i - \beta\eta \geq \Delta$ which implies that $\alpha_i \geq 0\ \forall i \in M \setminus J$. $\square$

Theorem 3.2 *Let $(\hat{z}, \hat{\eta}) \in \mathbb{R}^M \times \mathbb{R}$ such that $0 \leq \hat{z} \leq 1$ and $\sum_{i \in M, v_i = 0} \hat{z}_i \geq 1\ \forall v \in I$. $(\hat{z}, \hat{\eta}) \in P$ if and only if the optimal value of the following linear program is non-negative.*

$$
\begin{array}{lll}
\min_{(\alpha,\beta,\Delta)} & \alpha\hat{z} - \beta\hat{\eta} - \Delta & \\
s.t & & \\
& \alpha z - \beta ln(F(z)) - \Delta \geq 0 \; z \in S & \\
& \sum_{i\in M\setminus J} \alpha_i + \beta = 1 & \\
& \alpha_i \geq 0 \; i \in M\setminus J & \\
& \alpha_i = 0 \; i \in J & \\
& \beta \geq 0 &
\end{array} \tag{3.7}
$$

Furthermore, if (α, β, Δ) *is a feasible solution to* (3.7) *satisfying* $\alpha\hat{z} - \beta\hat{\eta} - \Delta < 0$, *then* $\alpha z - \beta\eta \geq \Delta$ *is a valid inequality for* P *which cuts off* $(\hat{z}, \hat{\eta})$.

Proof Clearly, if (α, β, Δ) is a feasible solution to (3.7), then $\alpha z - \beta\eta \geq \Delta$ is a valid inequality for P; hence if $(\hat{z}, \hat{\eta}) \in P$ then the optimal value of (3.7) is non-negative. Conversely, suppose $(\hat{z}, \hat{\eta}) \notin P$. For $i \in J$, $e - e^i$ is a p-inefficient point of F (Lemma 3.1) and hence $\hat{z}_i = 1$. Consequently, there exists a facet defining inequality $\alpha z - \beta\eta \geq \Delta$ of P which cuts off $(\hat{z}, \hat{\eta})$. Without loss of generality, we can assume that $\alpha_i = 0 \; \forall i \in J$. Since $0 \leq \hat{z} \leq 1$, the facet defined by $\alpha z - \beta\eta \geq \Delta$ is different from the ones defined by $z_i \leq 1 \; i \in M\setminus J$, which implies that $\alpha_i \geq 0 \; \forall i \in M\setminus J$, $\beta \geq 0$ and $\sum_{i\in M\setminus J} \alpha_i + \beta > 0$ (Lemma 3.1). If $\theta = \sum_{i\in M\setminus J} \alpha_i + \beta$, then $\frac{1}{\theta}(\alpha, \beta, \Delta)$ is feasible solution of P, $\frac{1}{\theta}(\alpha\hat{z} - \beta\hat{\eta} - \Delta) < 0$ and the optimal value of (3.7) is negative. □

Several comments are in order. First, the above theorem yields a systematic procedure for iteratively strengthening the (MIP1) formulation by generating cutting planes which cut off the incumbent fractional solution in each iteration. Furthermore, unlike the coefficient tightening procedure, the above separation procedure is guaranteed to produce every valid (facet-defining) inequality of P. Cuts derived using the separation linear program (3.7) are referred to as *polarity cuts* in the sequel.

Second, the linear program (3.7) has many more constraints than the number of variables, which suggests that the dual simplex algorithm is the most suitable linear programming algorithm for solving (3.7); the associated basis is a $(m + 1) \times (m + 1)$ matrix. Thus for $m = 10$, the dual simplex method updates the inverse of a 11×11 basis matrix. Third, we introduce a penalty term $\sum_{i\in M} w\alpha_i$ in the objective function, where $w = 10^{-4}$, which is aimed at favoring sparse cuts over equally good dense cuts (see Fischetti and Lodi [13] and de Souza and Balas [24] for importance of sparse cuts in cutting plane procedures).

Fourth, note that $(\hat{z}, \hat{\eta}) \in P$ if and only if $(e - \hat{z}, \hat{\eta}) \in \bar{P}$ where $\bar{P} = \{(z, \eta) \mid (e - z, \eta) \in P\}$ and e is a vector of ones. In other words, we can apply an affine transformation $(z, \eta) \mapsto (e - z, \eta)$ to $(\hat{z}, \hat{\eta})$, solve the separation linear program (3.7) in the transformed space and apply the inverse transformation to the cut (if any). The advantage of such a transformation is the reduction in the number of non-zeros in the coefficient matrix of (3.7) thereby improving the overall performance of the dual simplex algorithm due to sparsity considerations. To see this, note that 0-1 points $z \in S$ have significantly more number of ones than zeros. Figure 2 illustrates this phenomenon for Circular and Star distributions graphically. The horizontal axis represents the threshold probability p. The vertical axis represents the ratio of the number of non-zeros in the separation linear program formulated in the transformed space and original space, respectively, averaged over 1,000 randomly generated instantiations of

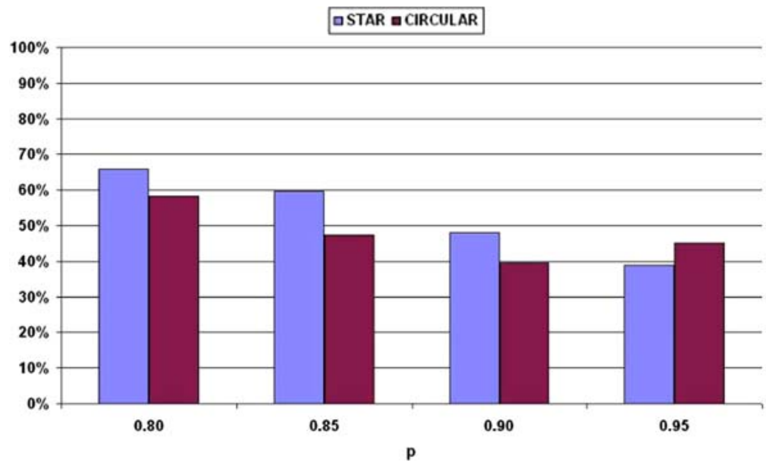

Fig. 2 Reduction in density of the coefficient matrix of separation linear program (3.7)

each one of the distributions (block size = 10). As is evident from Fig. 2, the above transformation can reduce the density of the coefficient matrix by $30 - 60\%$.

Finally, note that our separation linear program (3.7) bears a striking resemblance to the disjunctive programming based strengthening procedure proposed by Sen [23] in the early nineties. While both of our works draw motivation from the polarity theory, there is a minor difference in the precise application of the same. As in [5], Sen suggests computing the p-efficient frontier of the distribution of ξ, and then using the set of enumerated points to construct the polar linear program. Our approach, on the other hand, constructs the polar linear program for each individual block thereby circumventing the expensive enumeration of the p-efficient frontier. Indeed, for the special case when $L = 1$ both of these approaches are identical and derive the same cuts. To the best of our knowledge, the approach of Sen [23] was never computationally evaluated; the results presented in Sect. 5 highlight the tremendous computational utility of this approach.

Next we illustrate Theorem 3.2 on example (2.1) introduced in Sect. 2. The separation linear program for this example in the original and transformed space is given by,

$$
\underbrace{\left[\begin{array}{ll}
\min & \sum_{i=1}^{5} \alpha_i \hat{z}_i - \beta\hat{\eta} - \Delta \\
s.t & \\
& \alpha_3 + \alpha_4 + 0.04424\beta - \Delta \geq 0 \\
& \alpha_3 + \alpha_4 + \alpha_5 + 0.03318\beta - \Delta \geq 0 \\
& \alpha_2 + \alpha_3 + \alpha_4 + 0.03318\beta - \Delta \geq 0 \\
& \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + 0.02212\beta - \Delta \geq 0 \\
& \alpha_1 + \alpha_3 + \alpha_4 + 0.04424\beta - \Delta \geq 0 \\
& \alpha_1 + \alpha_3 + \alpha_4 + \alpha_5 + 0.02212\beta - \Delta \geq 0 \\
& \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + 0.02212\beta - \Delta \geq 0 \\
& \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 - \Delta \geq 0 \\
& \alpha_1 + \alpha_2 + \alpha_5 + \beta = 1 \\
& \alpha_1, \alpha_2, \alpha_5, \beta \geq 0 \\
& \alpha_3, \alpha_4 = 0
\end{array}\right]}_{\text{Original Space}}
\quad
\underbrace{\left[\begin{array}{ll}
\min & \sum_{i=1}^{5} -\alpha_i(1 - \hat{z}_i) - \beta\hat{\eta} - \Delta \\
s.t & \\
& -\alpha_1 - \alpha_2 - \alpha_5 + 0.04424\beta - \Delta \geq 0 \\
& -\alpha_1 - \alpha_2 + 0.03318\beta - \Delta \geq 0 \\
& -\alpha_1 - \alpha_5 + 0.03318\beta - \Delta \geq 0 \\
& -\alpha_1 + 0.02212\beta - \Delta \geq 0 \\
& -\alpha_2 - \alpha_5 + 0.04424\beta - \Delta \geq 0 \\
& -\alpha_2 + 0.02212\beta - \Delta \geq 0 \\
& -\alpha_5 + 0.02212\beta - \Delta \geq 0 \\
& -\Delta \geq 0 \\
& \alpha_1 + \alpha_2 + \alpha_5 + \beta = 1 \\
& \alpha_1, \alpha_2, \alpha_5, \beta \geq 0 \\
& \alpha_3, \alpha_4 = 0
\end{array}\right]}_{\text{Transformed Space}}
$$

The proposition that follows gives an alternative MIP reformulation of (PSC) using the apparatus of polarity cuts. The assumption that $L = 1$ is no longer made in the following discussion. For $t = 1 \dots L$, let

$$P_t = \text{clconv } \{(v, \eta) \mid v \in S_t,\ \eta \leq ln(F_t(v))\}.$$

Proposition 3.3 [22] (PSC) *can be reformulated as the following mixed integer program.*

$$\min_{(x,z,\eta)} \left\{ cx \;\middle|\; \begin{array}{c} Ax \geq z,\ 0 \leq x, z \leq 1 \\ \sum_{t=1}^{L} \eta_t \geq ln(p) \\ (z^t, \eta_t) \in P_t\ \forall t = 1 \dots L \\ x \in \{0, 1\}^N \end{array} \right\} \tag{R2}$$

Some comments are in order. First, the constraint $(z^t, \eta_t) \in P_t$ in (R2) can be replaced by a system of inequalities which define P_t; these inequalities, in turn, can be separated efficiently using the separation algorithm discussed above.

Second, while (R2) and (MIP1') are both valid reformulations of (MIP1) containing no additional integer constrained variables, the LP relaxation of (R2) is much stronger than the LP relaxation of (MIP1'), as confirmed by our computational results (see Sect. 5). Besides, special properties of the distribution can at times be used to represent the condition $(z^t, \eta_t) \in P_t$ in (R2) compactly using a polynomial number of additional constraints and variables, yielding a formulation which can be used to address large scale problems. See Sect. 4 for an example of such a distribution.

Third, consider the case when all components of the random 0-1 vector ξ are pairwise independent. In other words, $L = m$ and $|M_t| = 1$ for $t = 1 \dots L$. If $M_t = \{i\}$ and $p_t = \mathbb{P}(\xi_i \leq 0)$ then

$$P_t = \begin{cases} \{(v, \eta) \mid 0 \leq v \leq 1, \eta \leq (ln\ p_t)(1 - v)\} & \text{if } p_t \geq p \\ \{(v, \eta) \mid v = 1, \eta \leq 0\} & \text{if } p_t < p \end{cases}$$

for $t = 1 \dots L$ (see Fig. 3). Consequently, in this case the (R2) relaxation is identical to (MIP1').

4 Stationary distributions

In the previous section, we demonstrated how structural properties of the matrix A can be used to devise reformulations of (MIP1) with fewer number of integer variables. In this section, we investigate the same question apropos of the probability distribution function F.

A cumulative distribution function $F : \{0, 1\}^M \to \mathbb{R}$ is said to be *stationary* if $F(v) = F(w)\ \forall v, w \in \{0, 1\}^M$ such that $\sum_{i \in M} v_i = \sum_{i \in M} w_i$. Thus the value $F(z)$ of a stationary distribution depends only on the number of ones in z. More precisely, any stationary distribution is completely defined by a vector $(\lambda_0, \dots, \lambda_m)$ $(m = |M|)$ where λ_i represents the value of the stationary distribution at a lattice point with

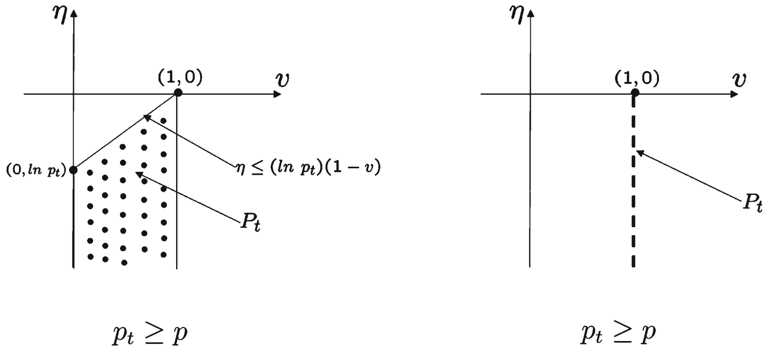

Fig. 3 P_t for the case when $|M_t| = 1$

exactly i ones. Note that if F is a stationary distribution then the corresponding random variables $\{\xi_i \mid i \in M\}$ are Exchangeable Random Variables which have been extensively studied in the literature (see [15]).

For the sake of an example, consider the vehicle routing problem [25] wherein the coverage requirements are modelled using the set covering constraints $Ax \ge 1$; the columns of the 0-1 matrix A represent the set of available routes, the rows of A represent the set of customers, and $A_{ij} = 1$ if and only if the jth route covers the ith customer, for all i, j. In the probabilistic variant of the problem, the constraints $Ax \ge 1$ are replaced by $\mathbb{P}(Ax \ge \xi) \ge p$ wherein the 0-1 random vector ξ models the variability in the set of realized customers. In many practical applications of vehicle routing such as food-delivery systems, coordination of limousine services, design of emergency logistic support systems, hazardous material distribution systems etc, the variability in the set of customers behaves as a *macro* property, i.e. given a number k, the probability that exactly k customers place a request for service is a function of k only, and is independent of the specific set of k customers who place the request. Alternatively, the distribution function of ξ depends only on the number of ones in the corresponding lattice point, and is hence a stationary distribution.

The proposition that follows gives a polynomial (in m) sized MIP reformulation of (MIP1) for the case of stationary distributions.

Proposition 4.1 *Suppose for $t \in \{1 \ldots L\}$, F_t is a stationary distribution defined by the vector $(p_0^t, p_1^t \ldots p_{m_t}^t)$ $(m_t = |M_t|)$ and $k_t = \min\{k \mid 0 \le k \le m_t,\ p_k^t \ge p\}$. (MIP1) is equivalent to*

$$
\begin{aligned}
\min_{(x,z,\eta,y,w)} \quad & cx \\
s.t \quad & \\
& Ax \ge z \\
& x_j \in \{0,1\}\ j \in N \qquad \text{(MIP2)} \\
& z_i \in \{0,1\}\ i \in M \\
& \textstyle\sum_{t=1}^{L} \eta_t \ge ln\ p
\end{aligned}
$$

$$\left.\begin{aligned} y_t &= \textstyle\sum_{i \in M_t} z_i \\ y_t &\geq k_t \\ y_t &\in \mathbb{Z} \end{aligned}\right\} \quad t = 1 \ldots L$$

$$\left.\begin{aligned} y_t + w_k^t(n-k) &\leq n \\ y_t + w_k^t(k+1) &\geq k+1 \\ \eta_t &\leq (ln\ p_k^t) w_k^t \\ w_k^t &\in \{0, 1\} \end{aligned}\right\} \quad k = 1 \ldots m_t,\ t = 1 \ldots L$$

Proof Suppose (x, z, η, y, w) is a feasible solution to (MIP2). Let $t \in \{1 \ldots L\}$. Note that $y_t = \sum_{i \in M_t} z_i$ is equal to the number of ones in z^t. Furthermore, $w_k^t = 1$ if and only if $y_t \leq k$ for $k = k_t \ldots m_t$. The constraint $y_t \geq k_t$ ensures that $\mathbb{P}(\xi_i \leq z_i \mid i \in M_t) \geq p$ and $z^t \in S_t$, whereas the constraint $\eta_t \leq (ln\ p_k^t) w_k^t$ ensures that $\eta_t \leq ln\ \mathbb{P}(\xi_i \leq z_i \mid i \in M_t)$. Hence (x, z, η) is a feasible solution to (MIP1). Conversely, suppose (x, z, η) is a feasible solution to (MIP1); (x, z, η, y, w) is a feasible solution to (MIP2) where y, w are defined as: for $t = 1 \ldots L,\ y_t = \sum_{i \in M_t} z_i$ and for $k = k_t \ldots m_t,\ w_k^t = 1$ if and only if $y_t \leq k$. □

Note that (MIP2) has linear (in m) number of additional variables and constraints, and hence can be used to handle arbitrarily large block sizes. Suppose $F : \{0, 1\}^M \to \mathbb{R}$ is a stationary cumulative distribution function defined by the vector $(p_0 \ldots p_m)$. It is worth observing that the lattice $\{0, 1\}^M$ associated with F can be partitioned into $(m+1)$ slices such that the k^{th} slices is composed of 0-1 M-vectors with exactly k ones for $k = 0, 1 \ldots m$, and the closed convex hull of the set $\{(z, \eta) \mid z \in \{0, 1\}^M,\ \sum_{i \in M} z_i = k,\ \eta \leq ln(F(z))\}$ has a compact description given by $\{(z, \eta) \mid 0 \leq z \leq 1,\ \sum_{i \in M} z_i = k,\ \eta \leq ln\ p_k\}$. In other words, stationary distributions posses the disjunctive shattering property defined below.

Definition 4.2 A cumulative distribution function $F : \{0, 1\}^M \to \mathbb{R}$ is said to posses the *disjunctive shattering property* (DSP) if the lattice $\{0, 1\}^M$ can be partitioned into polynomial (in m) number of subsets, say $\{0, 1\}^M = \cup_{j=1}^k M(j)$, such that the closed convex hull of the set $\{(z, \eta) \mid z \in M(j),\ \eta \leq ln(F(z))\}$ has a polynomial (in m) sized compact description $\{(z, \eta) \mid A^j z + d^j \eta \geq b^j\}$ for $j = 1 \ldots k$.

The proposition that follows gives a polynomial sized reformulation of (MIP1) for the case when each one of the distribution functions F_t possesses the disjunctive shattering property.

Proposition 4.3 *For $t \in \{1 \ldots L\}$, suppose F_t possesses the disjunctive shattering property and the lattice $\{0, 1\}^{M_t}$ corresponding to F_t is partitioned into k_t subsets, say $\{0, 1\}^{M_t} = \cup_{j=1}^{k_t} M_t(j)$, and for $j = 1 \ldots k_t$,* clconv $\{(v, \eta) \mid v \in M_t(j), \eta \leq ln\ F_t(v)\} = \{(v, \eta) \mid A^{jt} v + d^{jt} \eta \geq b^{jt}\}$. *(MIP1) is equivalent to,*

$$
\begin{array}{lll}
\min_{(x,z,\eta,\tilde{z},\tilde{\eta},\lambda)} & cx & \\
s.t & & \\
& Ax \geq z & (MIP3) \\
& x_j \in \{0,1\}\ j \in N & \\
& \sum_{t=1}^{L} \eta_t \geq ln\ p & \\
& \left.\begin{array}{r} z_i = \sum_{j=1}^{k_t} \tilde{z}_i^{jt}\ i \in M_t \\ \eta_t = \sum_{j=1}^{k_t} \tilde{\eta}^{jt} \\ \sum_{j=1}^{k_t} \lambda_{jt} = 1 \\ A^{jt}\tilde{z}^{jt} + d^{jt}\tilde{\eta}^{jt} \geq \lambda_{jt} b^{jt}\ j = 1 \ldots k_t \\ \lambda_{jt} \geq 0\ j = 1 \ldots k_t \end{array}\right\} t = 1 \ldots L &
\end{array}
$$

Proof Let $t \in \{1 \ldots L\}$. Note that $P_{jt} = \{(v, \eta) \mid A^{jt}v + d^{jt}\eta \geq b^{jt}\} \neq \emptyset\ \forall j = 1 \ldots k_t$, and the following constraints define the extended formulation (see Balas [1]) of $P_t = \text{clconv}\left(\cup_{j=1}^{k_t} P_{jt}\right)$ for $t = 1 \ldots L$.

$$z_i = \sum_{j=1}^{k_t} \tilde{z}_i^{jt}\ i \in M_t$$

$$\eta_t = \sum_{j=1}^{k_t} \tilde{\eta}^{jt}$$

$$\sum_{j=1}^{k_t} \lambda_{jt} = 1$$

$$A^{jt}\tilde{z}^{jt} + d^{jt}\tilde{\eta}^{jt} \geq \lambda_{jt} b^{jt}\ j = 1 \ldots k_t$$

$$\lambda_{jt} \geq 0\ j = 1 \ldots k_t$$

Consequently, (MIP2) is equivalent to (R2) and the above proposition follows immediately from Theorem 3.3. □

Note that (MIP3) has polynomial (in m) number of additional variables and constraints, and has no additional integer constrained variables. (MIP3) can be generalized to handle the case when some or all of the system of inequalities $A^{jt}z^{jt} + d^{jt}\eta^{jt} \geq b^{jt}$ have exponential number of inequalities, provided there exists a polynomial time separation algorithm to identify a violated inequality among $A^{jt}z^{jt} + d^{jt}\eta^{jt} \geq b^{jt}$. The generalization, however, is technical and is of limited interest in the context of the current paper.

5 Computational results

Figure 4 gives the flowchart of our algorithm to solve (PSC). We implemented our algorithm using COIN-OR [9] and CPLEX (version 9.0). The linear programming module (OsiClp) of COIN-OR was used to solve all resulting linear programs, while the final

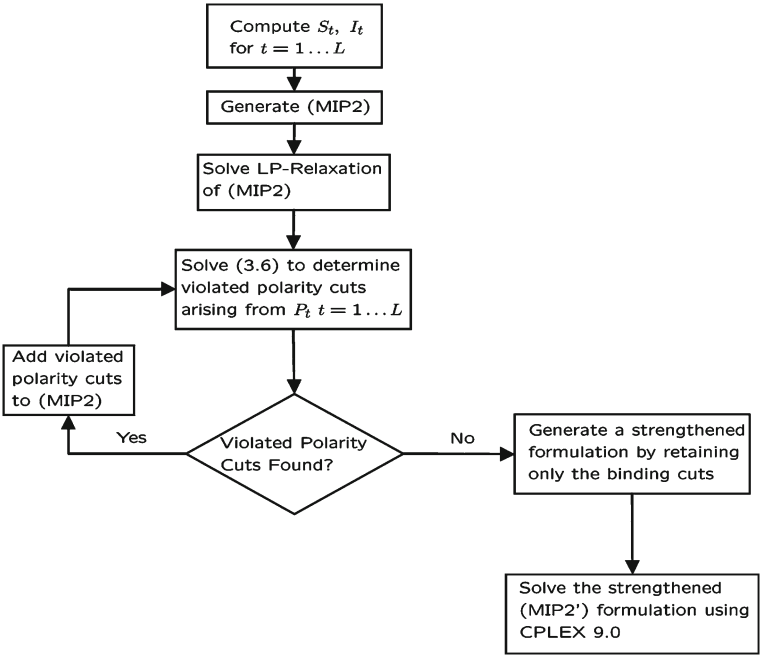

Fig. 4 Flow chart of the algorithm

MIP formulation was solved using CPLEX 9.0. All experiments were carried out on a 2 GHz P4 processor with 2 GB RAM. In this section, we describe the computational results of our experiment on a test-bed consisting of several thousand probabilistic instances.

Besides the probabilistic set covering instances, we also ran our code on probabilistic versions of the Single Source Capacitated Facility Location Problem (SSCFLP), Capacitated Warehouse Location Problem (CWLP) and Capacitated k-Median Problem (k-median) instances. We considered the following probabilistic version of SSCFLP.

$$\begin{array}{ll} \min_{(x,y)} \sum_{i\in I, j\in J} c_{ij}x_{ij} + \sum_{i\in I} f_i y_i & \\ s.t & \\ \qquad P\left(\sum_{i\in I} x_{ij} \geq \xi_i \ \forall j \in J\right) \geq p & \\ \qquad \sum_{j\in J} w_j x_{ij} \leq s_i y_i \ \forall i \in I & . \\ \qquad x_{ij} \in \{0,1\} \ \forall i \in I, \forall j \in J & \\ \qquad y_i \in \{0,1\} \ \forall i \in I & \end{array} \tag{5.8}$$

Here I is the set of facilities, s_i is the capacity and f_i is the fixed cost associated with facility $i \in I$, while J is the set of customers, w_j is the demand of customer j and c_{ij} is the cost of serving customer j via the facility i for all i, j. The model obtained by replacing the integrality constraints on the x_{ij} variables in (5.8) by $0 \leq x_{ij} \leq 1$ is the probabilistic version of the CWLP. Similarly, appending the constraint

$\sum_{i \in I} y_i \leq k$ ($k \geq 1$) to the probabilistic CWLP yields the probabilistic version of the k-median problem. Note that all of our results in Sect. 2 (except Proposition 2.3), 3 and 4 (except Proposition 4.3) can be applied to probabilistic CWLP and k-median problems too. Since these results form the basis of the our computational experiments, we decided to include the probabilistic variants of CWLP and k-median models in our test-bed.

Table 2 gives detailed information about the deterministic instances we chose from the literature, which were subsequently used to generate a test-bed of probabilistic instances as described below. From each problem set, we retained only those instances which could be solved to optimality by the default version of CPLEX 9.0 within a time-limit of 1 hr. Note that this selection criterion allows us to generate a test-bed of easy and moderately difficult instances which can be used to gain insights into the interplay between the integrality and probabilistic constraints of (PSC). Indeed, some extremely difficult set covering and SSCFLP instances were excluded by this selection criterion; these instances are likely to give rise to extremely difficult probabilistic instances wherein the integrality constraints of (PSC) themselves make the problem difficult to solve, their interaction with the probabilistic constraints notwithstanding.

From each deterministic instance we generated 20 probabilistic instances in the following manner. Following [5], we considered two different block sizes, namely 5 and 10. For each one of these block sizes, we considered two different probability distributions namely, Circular and Star (see [5] for the definition of these distributions). For sake of completeness, we also considered the case of independent random variables. In particular, we have assumed that each component ξ_i $(i = 1 \dots m)$ can take value 0 with probability $q_i = q_0^{1/i}$ where $0 < q_0 < 1$. Following [5], we used $q_0 = 0.1$ in our experiments. For each one of the five combinations of block sizes and distribution type, we generated four probabilistic problems differing only in the values of the threshold probabilities which were chosen from $\{0.80, 0.85, 0.90, 0.95\}$.

For the case of Set Covering and SSCFLP instances we used the (MIP1') formulation whereas for the case of CWLP and k-median instances we used the (MIP1) formulation. We strengthened the initial formulation by polarity cuts (Sect. 3) for instances which were generated using the Circular and Star distribution. Since (MIP1') formulation cannot be strengthened by polarity cuts for the case of independent distribution (see Sect. 3), the polarity cuts generator was turned off for these instances. For each probabilistic instance we ran our code with a time limit of 1 hr.

Tables 9–16 report our key findings. The first four tables give statistics on the performance of our algorithm while the last four tables give detailed statistics on the performance of polarity cuts. The results are categorized by distribution type, block size and threshold probability p which are given in the first, second and third columns of the tables respectively. Note that for each combination of distribution type, block size and threshold probability, the test-bed had 60 set covering instances, 70 SSCFLP instances, 37 CWLP instances and 20 k-median instances as reported in Table 2. Table 3 summarizes the computational results.

The fourth column in Tables 9–12 gives the number of instances which could not be solved to optimality within the prescribed time-limit of 1 hr. Of the 3,740 probabilistic instances on which we had run our code, we were able to solve 3,703 instances to optimality within 1 hr. In order to assess the performance of our algorithm over the

Table 2 Test bed of deterministic instances

Deterministic problem	Source	Total number of instances	Number of instances retained	Instances excluded	# Constraints	# Columns	# Probabilistic constraints
Set Covering	ORLIB [4]	80	60	scpclr 10–13	50–500	500–5000	50–500
				scpcyc 06–11			
				scpnrg 1–5			
				scpnrh 1–5			
SSCFLP	Holmberg [14]	71	70	p58	60–230	510–6030	50–200
CWLP	ORLIB [4] (Set 1)	37	37	–	66–100	816–2550	50
k-median	ORLIB [4]	20	20	–	101–201	2550–10100	50–100

Table 3 Summary results: circular, star and independent distributions

Problem class	# Probabilistic instances	# Unsolved instances	% RG	Solution time (sec)	# B& B nodes	Strengthening by polarity cuts	
						Time spent (sec)	% Duality gap closed
Set Covering	1200	21	10.89	153.59	6434.76	0.781	31.65
SSCFLP	1400	16	0.43	36.11	2488.26	0.190	27.67
CWLP	740	0	0.00	0.35	37.65	0.046	17.17
k-Median	400	0	0.00	50.88	1636.70	0.076	0.00

unsolved instances, we give the percentage relative gap[3] which remained at the end of 1 hr in the fifth column of the table, averaged over instances which could not be solved to optimality. The next two columns give the total solution time and the number of branch-and-bound nodes enumerated by CPLEX, averaged over instances which could be solved to optimality within 1 hr. The eigth column of the table gives the average value of the probabilistic information, calculated as $VOI = 100 \times \frac{Det(ip) - Prob(ip)}{Det(ip)}$, where VOI is the value of probabilistic information, Det(ip) is the optimal value of the deterministic problem and Prob(ip) is the optimal value of the probabilistic problem. Note that VOI represents the savings which could be made by incorporating the probabilistic information into the optimization model.

Tables 13–16 give detailed statistics on the performance of polarity cuts in our algorithmic framework. The fourth column of these tables reports the average number of rounds of polarity cuts which were generated by our code. The next column reports the average duality gap[4] closed at the root node after our code ceased to produce violated polarity cuts. The next two columns report the average time spent on strengthening the (MIP1) formulation by means of polarity cuts; the first column reports the total time spent on strengthening while the following column reports the time spent exclusively on solving the separation linear programs (3.7). Note that most of the time spent on strengthening was used to solve the LP relaxations of the (MIP1) formulation, and a very small fraction (less than 8% on average) was spent on solving the separation linear programs. This suggests that polarity cuts can be combined with other families of cutting planes such as Mixed Integer Gomory cuts or split cuts with very little computational overheads. The next two columns report the total number of cuts which were generated and the number of cuts which were binding at the final iteration, respectively, averaged over instances in the respective category.

The last four columns of Tables 13–16 report statistics on the fractionality of the optimal solution to the LP relaxation of the (MIP1) formulation before and after adding the polarity cuts. Given a feasible solution (x, z, η) to the (possibly strengthened) LP relaxation of (MIP1), let $f_z = |\{i \in M \mid 0 < z_i < 1|$ and $f_g = |\{t \in \{1 \ldots L\} \mid \exists i \in M_t \ s.t\ 0 < z_i < 1\}|$; f_z is a measure of fractionality of (x, z, η) in the z_i components whereas f_g measures the same in an aggregated form. The last four columns of the tables report the average values of f_z and f_g before and after adding the polarity cuts. Note that polarity cuts reduce the number of fractional z components by 80% on average. Furthermore, the impact of polarity cuts on the fractionality of the incumbent solution is more pronounced in the case of SSCFLP instances as compared to the set covering instances.

It is interesting to note that polarity cuts do not close any fraction of the duality gap on the k-median instances. This can be attributed to the specific structure of the k-median instances in the OrLib repository. These instances were generated by choosing random points in $[0, 100] \times [0, 100]$ where every point served as a customer and potential facility, and the cost of assigning a customer to a facility is the

[3] $RG = 100 \times \frac{ip - bb}{bb}$ where RG is the percentage relative gap, ip is the value of the best solution and bb is the value of the best bound available at the end of 1 hr

[4] $DG = 100 \times \frac{slp - lp}{ip - lp}$, where DG is the percentage duality gap closed, lp is the value of the LP relaxation of our model, slp is the value of the relaxation and ip is the value of the optimal solution.

Table 4 Characteristics of Beraldi and Ruszczyński's test problems

Problem	Distribution	Group size	Number of groups
Test 11	Star	5	40
Test 12	Circular	5	40
Test 13	Star	10	20
Test 14	Circular	10	20
Test 15	Independent	1	200

euclidean distance between the corresponding points, rounded down to the nearest integer. Consequently, the LP relaxation of these models fractionally assigns the customer at (x, y) to the facility at (x, y) thereby giving a relaxation value of 0. The same argument carries over to the relaxation of the probabilistic version strengthened by polarity cuts, thus explaining the zeros in column 5 of Table 16. Nevertheless, polarity cuts are indeed effective in decreasing the fractionality of the optimal LP relaxation solutions of these instances, as shown by the last 4 columns of Table 16.

Next we compare our results with the earlier work of Beraldi and Ruszczyński [6] who conducted their experiments on two set covering instances, namely scp41 and scp42, from the ORLIB repository [4]. Both of these problems have 200 set covering constraints. They constructed 20 probabilistic instances from these two instances by considering the five combinations of group sizes and distribution types shown in Table 4, and two values of the threshold probability p, namely 0.90 and 0.95, for each combination. They tested several variants of their algorithm and concluded that a certain variant, which they refer to as the hybrid strategy with simple heuristic, performs best on their test-bed.

Table 5 compares the performance of our algorithm with the best version of the algorithm proposed in [6] on the test-bed constructed in [6] (see [6] for description of these instances). The first column of the table gives the problem description; a suffix of 1 (2) indicates that the instance was generated from scp41 (scp42). The second column gives the value of threshold probability p. The next two columns report the computational results of [6]; the first column gives the total computing time while the following column gives the number of p-efficient points which were enumerated by their algorithm. The next two columns report the performance of our algorithm on these instances. The first column reports the total computing time while the following column gives the number of branch-and-bound nodes enumerated by CPLEX. Notice that the computing time of our approach is several orders of magnitude better than that of [6]. Furthermore, the extent of enumeration in our approach (column 6) is substantially smaller than in the approach proposed in [6] (column 4). It should, however, be stressed that our approach is tailored to exploit the block-structure of the random variable ξ whereas the approach of [6] is a general solution method which does not attempt to exploit special properties of the ξ. This observation partially explains the better performance of our approach on these instances.

Next we discuss the importance of polarity cuts in the overall solution procedure by demonstrating their impact on solving the probabilistic version of a SSCFLP instance. We chose the instance p31 (30 facilities and 150 customers) from the Holmberg test-bed [14] and generated its probabilistic variant using the Circular distribution and threshold probability $p = 0.8$; the resulting probabilistic instance had 15 blocks of size

Table 5 Comparison with the Beraldi and Ruszczyński algorithm

Problem	p	BR		SGL	
		Solution time (sec)	PEP	Solution time (sec)	# B&B nodes
Test 11.1	0.95	6.75	209	0.02	0
Test 11.1	0.90	60.87	935	0.03	0
Test 12.1	0.95	7.76	165	0.06	25
Test 12.1	0.90	448.52	6387	0.02	0
Test 13.1	0.95	3.99	79	0.04	5
Test 13.1	0.90	37.90	463	0.04	0
Test 14.1	0.95	8.83	252	0.02	0
Test 14.1	0.90	380.76	5284	0.02	0
Test 15.1	0.95	8780.02	140274	0.00	0
Test 15.1	0.90	33153.07	529814	0.02	0
Test 11.2	0.95	7.68	221	0.02	0
Test 11.2	0.90	123.41	1881	0.08	0
Test 12.2	0.95	4.02	106	0.06	0
Test 12.2	0.90	400.20	6249	0.04	0
Test 13.2	0.95	8.65	217	0.02	0
Test 13.2	0.90	155.64	1745	0.14	10
Test 14.2	0.95	13.32	297	0.18	27
Test 14.2	0.90	911.31	13267	0.13	6
Test 15.2	0.95	8170.48	130049	0.02	0
Test 15.2	0.90	24581.25	389886	0.02	0

10 each. We ran our code on this instance in three setups. In the first setup we ran the default version of our algorithm which involves, among other things, adding polarity cuts at the root node. The second and third setups were identical to the first setup, except that the generator for polarity cuts was turned off; we used the (MIP1) and (MIP1') formulations in the second and third setups, respectively. In the first setup, our code closed 67.84% of the duality gap at the root node in less than 1 s; CPLEX acting on the formulation, strengthened by polarity cuts, was able to solve the instance to optimality in additional 52 s by enumerating 2378 branch and bound nodes. The second setup involved applying CPLEX to the unstrengthened formulation. Interestingly CPLEX, unaided by the polarity cuts, was not able to solve the instance to optimality in 2 hr; it enumerated around 154,100 branch-and-bound nodes and closed only 72% of the duality gap at the end of two hours. After additional 31 hours CPLEX was able to solve the instance to optimality by enumerating 1.7 million branch-and-bound nodes. CPLEX took around 21 hrs to solve the instance to optimality in the third setup and enumerated 764,006 branch-and-bound nodes. Table 6 summarizes the statistics associated with these three setups. As this example demonstrates, polarity cuts have a huge impact on the overall solution time of our procedure.

An interesting question is to determine if the strengthening which results from the addition of polarity cuts can also obtained by adding general purpose cutting planes

Table 6 Probabilistic version of the SSCFLP instance p31

With Polarity Cuts (MIP1')	
% Gap closed at root node	67.84%
Time spent in strengthening	0.83 s
Time spent in solving SepLP	0.30 s
Time Taken by CPLEX 9.0 after Strengthening	52.31 s
No. of Branch-and-Bound nodes enumerated by CPLEX 9.0	2378
Total time taken to solve the instance to optimality	53.14 s
Without Polarity Cuts (MIP1')	
No. of Branch-and-Bound nodes enumerated by CPLEX 9.0	764,006
Total time taken to solve the instance to optimality	77621 s
Without Polarity Cuts (MIP1)	
No. of Branch-and-Bound nodes enumerated by CPLEX 9.0	1,717,126
Total time taken to solve the instance to optimality	119,922 s

such as mixed integer Gomory cuts or lift-and-project cuts to the (MIP1) formulation. In order to answer this question, we tried to strengthen the (MIP1) formulation of the probabilistic SSCFLP p31 instance (described above) by using other well-known classes of cutting planes. Table 7 summarizes our key findings. The first column of the table reports the type of cutting plane procedure; the second column reports the number of rounds of cuts which were generated. The third column gives the percentage duality gap closed by the respective class of cutting planes while the last column reports the total time spent on strengthening. The first row reports the performance of polarity cuts. The second row reports the performance of cuts generated by CPLEX 9.0 MIP solver at the root node; CPLEX was used in the "move best bound" mode and all of its cuts generators (except disjunctive cuts) were used in the "aggressive" mode so as to extract the best performance of the CPLEX cut generators. The remaining rows report the performance of mixed integer Gomory (MIG) cuts, mixed integer rounding (MIR) cuts, reduce-and-split (RedSplit) cuts and lift-and-project (L&P) cuts. We used the COIN-OR modules CglGomory, CglMixedIntegerRounding and CglRedSplit to generate MIG, MIR and reduce-and-split cuts, respectively; the lift-and-project cuts were generated with the code used in Balas and Saxena [2]. Note that among all classes of cutting planes, polarity cuts close the maximum fraction of the duality gap in minimum amount of time.

We conclude this section by reporting our computational experience with the (MIP4) formulation for stationary distributions. Recall that any stationary distribution is completely defined by a vector $(\lambda_0, \ldots, \lambda_m)$ $(m = |M|)$ where λ_i represents the value of the stationary distribution at a lattice point with exactly i ones. We used the following scheme to generate m-dimensional stationary distributions.

1. Let $a_j = \frac{100u_j}{2^{j+1}}$ $j = 0 \ldots m$ where u_j is a random number in the interval (0, 1).
2. Let $\mu = \sum_{j=0}^{m} a_j$ and let $a_j := \frac{a_j}{\mu}$ $j = 0 \ldots m$.
3. For $j = 0 \ldots m$, let $\lambda_j = \sum_{k=0}^{j} a_k \Pi_{i=0}^{k-1} \frac{j-i}{m-i}$.

Table 7 Comparing polarity cuts with other general purpose cutting planes

Type of cuts	# Rounds	% Duality gap closed	Time (sec)
Polarity Cuts	5	67.84	0.83
CPLEX	–	48.35	16.87
RedSplit	5	8.66	4.27
	10	9.99	12.50
MIR	5	11.42	0.67
	10	12.21	1.31
MIG	5	26.07	0.40
	10	31.71	1.09
	15	34.94	1.19
	20	36.45	2.18
L&P	5	38.03	37.72
	10	44.81	95.09
	15	48.15	177.45
	20	50.01	292.46

Table 8 Summary results: stationary distribution

Problem class	# Probabilistic instances	# Unsolved instances	% RG	Solution time (sec)	# B& B nodes
Set Covering	1920	128	21.34	112.93	9373.60
SSCFLP	2240	17	0.35	9.36	1609.35
CWLP	1184	0	0.00	0.09	5.25
k-Median	640	0	0.00	2.90	156.22

It can be easily verified that $(\lambda_0, \ldots, \lambda_m)$ obtained by the above procedure defines a cumulative distribution function of a m-dimensional stationary distribution.

For each deterministic instance (Table 2) we generated 32 probabilistic instances in the following manner. We considered 8 different block sizes, namely 5, 10, 20, 50, $\frac{m}{4}$, $\frac{m}{3}$, $\frac{m}{2}$ and m where m denotes the number of probabilistic set covering constraints in the deterministic instance. For each block, the stationary distribution was defined using the scheme described above. For each one of these block sizes we generated four probabilistic problems differing only in the values of the threshold probabilities which were chosen from $\{0.80, 0.85, 0.90, 0.95\}$. We solved each one of the resulting probabilistic instance by CPLEX 10.1 with a time limit of 1 hr.

Table 8 summarizes our computational results. Our goal in this experiment was to verify whether special properties of distributions can be exploited to solve probabilistic problems with arbitrarily large block sizes. As is evident from Table 8 our goal was largely attained, at least on this test-bed of problem instances.

6 Concluding remarks

In this paper, we set out to explore MIP reformulations of the probabilistic set covering problem (PSC) which exploit the block structure of the random variable ξ.

We introduce the concepts of p-inefficiency and polarity cuts. While the former is aimed at reducing the number of constraints in our model, the latter is used as a strengthening device to obtain stronger formulations. Simplifications of the MIP model which result due to special properties of matrix A and distribution function F are briefly discussed. We corroborate our theoretical findings by an extensive computational experiment on a test-bed consisting of almost 10,000 probabilistic instances. Tables 3 and 8 summarize our computational results.

This paper treads on the interface of two important areas of computational optimization—probabilistic programming and mixed integer programming. The main contribution of the paper, however, lies in integrating celebrated concepts from each one of these fields, namely p-efficiency from probabilistic programming and polarity from mixed integer programming, to create an algorithmic framework to solve (PSC) which is orders of magnitude more efficient than any of the existing approaches.

Acknowledgments The authors will like to thank anonymous referees for their useful comments and for bringing the work of Sen [23] to our knowledge.

Appendix

Table 9 Summary results: set covering instances

Distribution	Block size	p	# Unsolved instances	% RG	Solution time (sec)	# B&B nodes	% VOI
Circular	5	0.80	3	10.41	91.30	3736	9.66
Circular	5	0.85	2	11.81	136.67	5943	7.17
Circular	5	0.90	1	11.34	176.73	8075	4.77
Circular	5	0.95	0	0.00	152.14	8820	2.11
Circular	10	0.80	2	9.94	185.26	5067	10.17
Circular	10	0.85	1	6.63	199.52	8059	7.45
Circular	10	0.90	1	9.36	218.97	10524	5.05
Circular	10	0.95	0	0.00	170.84	9946	2.55
Star	5	0.80	0	0.00	128.08	6075	11.91
Star	5	0.85	2	9.90	81.64	4209	9.98
Star	5	0.90	0	0.00	195.51	11825	7.02
Star	5	0.95	0	0.00	133.14	8836	3.44
Star	10	0.80	2	10.96	262.00	4124	14.51
Star	10	0.85	2	16.18	208.82	3787	12.19
Star	10	0.90	2	12.13	142.89	4618	9.44
Star	10	0.95	1	4.81	178.40	9091	4.69
Independent	1	0.80	0	0.00	91.04	2829	27.55
Independent	1	0.85	0	0.00	79.86	2677	22.15
Independent	1	0.90	0	0.00	132.02	5199	16.31
Independent	1	0.95	2	11.73	108.14	4873	10.14

Table 10 Summary results: SSCFLP instances

Distribution	Block size	p	# Unsolved instances	% RG	Solution time (sec)	# B&B nodes	% VOI
Circular	5	0.80	0	0.00	50.48	2599	8.88
Circular	5	0.85	0	0.00	29.02	1675	6.51
Circular	5	0.90	2	0.21	23.26	1779	4.21
Circular	5	0.95	1	0.85	45.21	4001	2.10
Circular	10	0.80	1	0.26	26.16	1808	7.15
Circular	10	0.85	0	0.00	72.43	2626	5.44
Circular	10	0.90	1	0.23	61.03	3574	3.40
Circular	10	0.95	0	0.00	35.79	3399	1.54
Star	5	0.80	0	0.00	30.38	1996	10.15
Star	5	0.85	0	0.00	16.22	1073	7.76
Star	5	0.90	0	0.00	11.88	963	5.36
Star	5	0.95	1	1.18	25.87	2174	2.84
Star	10	0.80	6	0.61	57.58	3054	12.64
Star	10	0.85	0	0.00	49.17	1858	9.74
Star	10	0.90	0	0.00	18.44	1028	6.73
Star	10	0.95	0	0.00	35.09	2134	3.47
Independent	1	0.80	2	0.09	58.97	6325	10.35
Independent	1	0.85	1	0.15	25.65	2497	7.90
Independent	1	0.90	1	0.01	9.84	1016	5.48
Independent	1	0.95	0	0.00	41.58	4326	2.75

Table 11 Summary results: CWLP instances

Distribution	Block size	p	# Unsolved instances	% RG	Solution time (sec)	# B&B nodes	% VOI
Circular	5	0.80	0	0.00	0.35	50	35.69
Circular	5	0.85	0	0.00	0.25	18	32.76
Circular	5	0.90	0	0.00	0.27	22	19.61
Circular	5	0.95	0	0.00	0.17	6	17.43
Circular	10	0.80	0	0.00	0.61	74	31.35
Circular	10	0.85	0	0.00	0.40	26	26.49
Circular	10	0.90	0	0.00	0.32	13	15.71
Circular	10	0.95	0	0.00	0.17	2	8.65
Star	5	0.80	0	0.00	0.49	89	32.71
Star	5	0.85	0	0.00	0.35	42	29.51
Star	5	0.90	0	0.00	0.38	48	25.84

Table 11 continued

Distribution	Block size	p	# Unsolved instances	% RG	Solution time (sec)	# B&B nodes	% VOI
Star	5	0.95	0	0.00	0.19	5	21.73
Star	10	0.80	0	0.00	0.90	157	15.70
Star	10	0.85	0	0.00	0.54	50	12.59
Star	10	0.90	0	0.00	0.47	62	8.78
Star	10	0.95	0	0.00	0.23	16	4.32
Independent	1	0.80	0	0.00	0.29	33	43.71
Independent	1	0.85	0	0.00	0.25	25	34.05
Independent	1	0.90	0	0.00	0.23	11	19.54
Independent	1	0.95	0	0.00	0.17	4	9.24

Table 12 Summary results: capacitated k-median instances

Distribution	Block size	p	# Unsolved instances	% RG	Solution time (sec)	# B&B nodes	% VOI
Circular	5	0.80	0	0.00	55.28	1822	15.11
Circular	5	0.85	0	0.00	32.01	948	11.56
Circular	5	0.90	0	0.00	28.96	1192	7.88
Circular	5	0.95	0	0.00	19.22	1007	3.89
Circular	10	0.80	0	0.00	125.50	3864	16.67
Circular	10	0.85	0	0.00	38.12	1210	12.44
Circular	10	0.90	0	0.00	19.08	524	8.31
Circular	10	0.95	0	0.00	32.15	2189	3.52
Star	5	0.80	0	0.00	86.11	2963	21.24
Star	5	0.85	0	0.00	86.72	2669	16.96
Star	5	0.90	0	0.00	33.70	871	12.40
Star	5	0.95	0	0.00	13.89	410	6.78
Star	10	0.80	0	0.00	212.07	5850	24.64
Star	10	0.85	0	0.00	125.58	3195	19.38
Star	10	0.90	0	0.00	44.88	1558	13.70
Star	10	0.95	0	0.00	17.44	457	7.00
Independent	1	0.80	0	0.00	14.38	708	16.34
Independent	1	0.85	0	0.00	10.36	441	12.80
Independent	1	0.90	0	0.00	11.79	458	8.95
Independent	1	0.95	0	0.00	10.31	398	4.68

Table 13 Performance of polarity cuts: set covering instances

Distribution	Block size	p	# Rounds	% Gap closed	Time (sec)		# Cuts		f_g		f_z	
					Total	SepLP	Total	Binding	Initial	Final	Initial	Final
Circular	5	0.80	4.6	32.55	0.706	0.012	17.7	14.8	10.2	7.8	29.7	10.1
Circular	5	0.85	4.7	28.90	0.678	0.009	14.4	12.2	8.1	6.3	23.7	7.9
Circular	5	0.90	4.1	20.90	0.584	0.006	9.7	8.0	5.8	4.7	16.6	5.7
Circular	5	0.95	2.5	9.30	0.345	0.003	2.9	2.6	3.9	3.5	6.9	3.8
Circular	10	0.80	5.7	40.44	0.976	0.033	22.5	17.8	10.4	7.5	47.6	11.0
Circular	10	0.85	5.8	38.45	0.842	0.022	19.3	15.3	8.5	6.1	38.3	8.6
Circular	10	0.90	6.0	30.91	0.731	0.013	14.8	11.1	6.4	4.7	28.3	6.5
Circular	10	0.95	3.7	16.27	0.392	0.003	5.2	4.2	3.8	3.2	10.0	3.9
Star	5	0.80	4.2	31.76	0.654	0.015	19.9	15.9	12.7	9.0	36.7	12.1
Star	5	0.85	4.0	30.52	0.642	0.011	16.0	12.9	10.2	7.8	29.5	10.4
Star	5	0.90	4.2	27.47	0.597	0.009	13.4	10.7	7.8	6.1	22.4	7.9
Star	5	0.95	3.0	12.81	0.474	0.003	6.2	5.4	4.9	4.1	12.2	4.7
Star	10	0.80	6.1	53.23	1.652	0.083	29.8	23.3	14.7	9.8	84.9	16.1
Star	10	0.85	6.1	51.18	1.384	0.061	25.6	20.0	12.6	8.4	72.2	13.3
Star	10	0.90	6.4	46.62	1.094	0.034	23.2	17.6	9.8	6.5	55.4	9.7
Star	10	0.95	5.7	35.08	0.748	0.013	16.5	12.1	6.3	4.4	34.0	5.9

Table 14 Performance of polarity cuts: SSCFLP instances

Distribution	Block size	p	# Rounds	% Gap closed	Time (sec)		# Cuts		f_g		f_z	
					Total	SepLP	Total	Binding	Initial	Final	Initial	Final
Circular	5	0.80	7.6	30.71	0.123	0.011	21.9	21.0	6.5	1.0	27.5	3.3
Circular	5	0.85	7.7	27.79	0.121	0.009	20.2	18.6	5.3	1.1	23.4	3.2
Circular	5	0.90	10.0	23.04	0.135	0.009	21.4	17.7	4.3	1.0	19.3	1.8
Circular	5	0.95	3.5	4.66	0.046	0.002	4.8	4.2	2.8	1.0	6.1	1.1
Circular	10	0.80	9.6	33.16	0.218	0.053	22.6	19.5	6.1	1.0	28.8	1.7
Circular	10	0.85	10.2	29.59	0.171	0.024	24.7	20.0	5.4	1.1	25.3	1.9
Circular	10	0.90	9.9	23.61	0.140	0.015	23.0	16.1	4.1	1.1	19.8	1.8
Circular	10	0.95	3.8	6.65	0.052	0.003	5.1	4.2	2.3	1.0	6.4	1.1
Star	5	0.80	5.9	31.83	0.106	0.010	18.4	17.7	8.8	1.1	31.1	1.6
Star	5	0.85	6.4	29.15	0.102	0.009	17.9	17.0	7.1	1.1	25.6	1.6
Star	5	0.90	7.8	24.07	0.108	0.008	17.9	16.1	5.5	1.0	19.6	1.6
Star	5	0.95	6.2	12.43	0.078	0.004	11.1	10.0	4.0	1.1	11.3	1.4
Star	10	0.80	17.8	49.29	0.589	0.128	42.6	27.5	7.7	1.1	52.0	2.4
Star	10	0.85	20.4	46.92	0.550	0.113	54.5	33.6	6.8	1.1	46.5	2.2
Star	10	0.90	18.7	41.72	0.346	0.057	48.7	30.1	6.0	1.1	41.7	1.5
Star	10	0.95	12.9	28.07	0.161	0.018	30.8	20.1	4.4	1.1	27.2	1.4

Table 15 Performance of polarity cuts: CWLP instances

Distribution	Block size	p	# Rounds	% Gap closed	Time (sec)		# Cuts		f_g		f_z	
					Total	SepLP	Total	Binding	Initial	Final	Initial	Final
Circular	5	0.80	5.0	6.35	0.028	0.002	5.5	5.2	2.5	1.0	5.6	2.5
Circular	5	0.85	3.6	3.79	0.025	0.001	3.4	3.2	2.0	1.0	3.7	1.3
Circular	5	0.90	2.0	1.28	0.014	0.001	1.1	1.1	1.1	1.0	2.5	1.0
Circular	5	0.95	1.1	0.06	0.009	0.000	0.1	0.1	1.1	1.0	1.3	1.0
Circular	10	0.80	7.8	33.78	0.111	0.031	17.6	15.1	4.0	1.1	22.8	1.6
Circular	10	0.85	7.5	29.34	0.071	0.012	12.3	10.6	3.2	1.0	18.1	1.5
Circular	10	0.90	7.5	14.41	0.051	0.006	10.7	7.6	2.2	1.0	13.5	1.0
Circular	10	0.95	2.4	1.20	0.015	0.001	1.4	1.4	1.1	1.0	2.1	1.0
Star	5	0.80	4.5	21.99	0.033	0.004	8.1	7.5	4.7	1.1	15.4	1.2
Star	5	0.85	4.2	24.57	0.035	0.004	7.0	6.6	4.7	1.0	14.3	1.4
Star	5	0.90	6.6	24.49	0.045	0.005	10.8	9.1	3.9	1.1	12.3	1.8
Star	5	0.95	2.9	5.73	0.021	0.002	4.0	3.7	3.0	1.0	6.4	1.1
Star	10	0.80	9.7	38.15	0.102	0.020	15.0	12.7	4.1	1.1	24.0	1.8
Star	10	0.85	11.5	35.02	0.096	0.015	18.7	13.2	4.1	1.2	22.9	2.9
Star	10	0.90	5.8	25.70	0.046	0.008	10.3	8.0	3.9	1.1	18.7	2.0
Star	10	0.95	6.1	8.81	0.039	0.004	8.1	6.7	2.1	1.0	10.4	1.0

Table 16 Performance of polarity cuts: capacitated k-median instances

Distribution	Block size	p	# Rounds	% Gap closed	Time (sec)		# Cuts		f_g		f_z	
					Total	SepLP	Total	Binding	Initial	Final	Initial	Final
Circular	5	0.80	2.6	0.00	0.058	0.002	4.6	4.5	4.1	0.7	6.8	0.8
Circular	5	0.85	2.5	0.00	0.055	0.001	3.7	3.7	3.3	0.6	5.8	0.6
Circular	5	0.90	2.5	0.00	0.053	0.001	2.8	2.7	2.6	0.9	4.3	1.1
Circular	5	0.95	2.2	0.00	0.036	0.000	1.5	1.5	1.4	0.5	2.4	0.5
Circular	10	0.80	3.1	0.00	0.103	0.010	6.8	6.5	4.7	0.5	20.6	0.5
Circular	10	0.85	4.0	0.00	0.097	0.007	6.7	5.8	3.8	0.7	17.2	0.8
Circular	10	0.90	4.5	0.00	0.075	0.004	6.0	4.2	2.8	0.9	10.3	1.0
Circular	10	0.95	2.1	0.00	0.037	0.000	1.1	1.0	1.1	0.5	2.6	0.5
Star	5	0.80	2.3	0.00	0.062	0.003	6.3	6.1	6.1	1.0	13.5	1.0
Star	5	0.85	2.1	0.00	0.055	0.002	5.5	5.5	5.5	0.9	11.7	0.9
Star	5	0.90	2.4	0.00	0.056	0.001	4.9	4.7	4.4	0.6	9.6	0.8
Star	5	0.95	2.4	0.00	0.045	0.001	2.5	2.4	2.3	0.7	4.2	0.7
Star	10	0.80	2.4	0.00	0.151	0.031	5.4	5.1	5.0	1.0	28.7	1.1
Star	10	0.85	3.3	0.00	0.141	0.024	6.2	5.1	4.6	1.0	27.3	1.0
Star	10	0.90	4.2	0.00	0.112	0.012	7.3	5.4	4.0	0.8	24.0	0.8
Star	10	0.95	4.8	0.00	0.084	0.005	6.4	4.6	2.8	0.9	11.8	1.1

References

1. Balas, E.: Disjunctive programming, properties of the convex hull of feasible points. Discret. Appl. Math. **89**(1-3), 3–44 (1998)
2. Balas, E., Saxena, A.: Optimizing over the split closure. Math. Program. Ser. A **113**(2), 219–240 (2008)
3. Bartholdi, J.J.III, Orlin, J.B., Ratliff, H.D.: Cyclic scheduling via integer programs with circular ones. Oper. Res. **28**, 1074–1085 (1980)
4. Beasley, J.E.: OR-Library. http://www.people.brunel.ac.uk/~mastjjb/jeb/info.html
5. Beraldi, P., Ruszczyński, A.: The probabilistic set covering problem. Oper. Res. **50**, 956–967 (2002)
6. Beraldi, P., Ruszczyński, A.: A branch and bound method for stochastic integer problems under probabilistic constraints. Optim. Methods Softw. **17**(3), 359–382 (2002)
7. Charnes, A., Cooper, W.W., Symonds, G.H.: Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. Manag. Sci. **4**, 235–263 (1958)
8. Christof, T., Lobel, A.: PORTA - POlyhedron Representation Transformation Algorithm. http://www.zib.de/Optimization/Software/Porta/
9. COIN: Computational infrastructure for operations research. http://www.coin-or.org
10. Cornuejols, G., Conforti, M.: Balanced 0,1,-1 Matrices, bicoloring and total dual integrality. Math. Program. **71**, 249–258 (1995)
11. Cornuejols, G.: Combinatorial Optimization: Packing and Covering. Published by SIAM (2001) in the CBMS-NSF Regional Conference Series in Applied Mathematics CBMS 74 (2000)
12. Dentcheva, D., Prékopa, A., Ruszczyński, A.: Concavity and efficient points of discrete distributions in probabilistic programming. Math. Program. **89**, 55–77 (2000)
13. Lodi, A., Fischetti, M.: Optimizing over the first Chvátal closure. Math. Program. **110**(1), 3–20 (2007)
14. Holmberg, K., Ronnqvist, M., Yuan, D.: An exact algorithm for the capacited facility location 517 problems with single sourcing. Eur. J. Oper. Res. **113**, 544–559 (1999)
15. Kallenberg, O.: Probabilistic Symmetries and Invariance Principles. Springer, Heidelberg (2005)
16. Lejeune, M.A., Ruszczyński, A.: An efficient method trajectory method for probabilistic inventory-production-distribution problems. Oper. Res. **55**(2), 378–394 (2007)
17. Luedtke, J., Ahmed, S., Nemhauser, G.: Strong MIP Formulations for Chance Constrained Linear Programs with Random Right-Hand Side. INFORMS Conference, Pittsburgh (2006)
18. Prékopa, A.: Dual method for a one-stage stochastic programming with random rhs obeying a discrete probability distribution. Z. Oper. Res. **34**, 441–461 (1990)
19. Prékopa, A.: Stochastic Programming. Kluwer, Boston (1995)
20. Prékopa, A.: Probabilistic programming models. In: Ruszczyński, A., Shapiro, A. (eds.) Stochastic Programming: Handbook in Operations Research and Management Science, vol. 10, Chap. 5, pp. 267–351. Elsevier Science Ltd, Amsterdam (2003)
21. Prékopa, A., Vizvari, B., Badics, T.: Programming under probabilistic constraint with discrete random variable. In: Giannessi, F., Komlósi, S., Rapcsák, T. (eds.) New Trends in Mathematical Programming. Boston, MA (1998)
22. Saxena, A., Goyal, V., Lejeune, M.A.: MIP reformulations of the probabilistic set covering problem, Tepper Working Paper No. 2007-E3 (2007)
23. Sen, S.: Relaxations for probabilistically constrained programs with discrete random variables. Oper. Res. Lett. **11**(2), 81–86 (1992)
24. de Souza, C., Balas, E.: The vertex separator problem: algorithms and computations. Math. Program. **103**(3), 609–631 (2005)
25. Toth, P., Vigo, D.: Branch-and-bound algorithms for the capacitated VRP. In: Toth, P., Vigo, D. (eds.) The Vehicle Routing Problem, pp. 29–52. SIAM, Philadelphia (2001)