



Sparse multi-term disjunctive cuts for the epigraph of a function of binary variables

Rui Chen¹ · James Luedtke²

Received: 11 July 2022 / Accepted: 14 August 2023

© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2023

Abstract

We propose a new method for separating valid inequalities for the epigraph of a function of binary variables. The proposed inequalities are disjunctive cuts defined by disjunctive terms obtained by enumerating a subset I of the binary variables. We show that by restricting the support of the cut to the same set of variables I , a cut can be obtained by solving a linear program with $2^{|I|}$ constraints. While this limits the size of the set I used to define the multi-term disjunction, the procedure enables generation of multi-term disjunctive cuts using far more terms than existing approaches. We present two approaches for choosing the subset of variables. Experience on three MILP problems with block diagonal structure using $|I|$ up to size 10 indicates the sparse cuts can often close nearly as much gap as the multi-term disjunctive cuts without this restriction and in a fraction of the time. We also find that including these cuts within a cut-and-branch solution method for these MILP problems leads to significant reductions in solution time or ending optimality gap for instances that were not solved within the time limit. Finally, we describe how the proposed approach can be adapted to optimally “tilt” a given valid inequality by modifying the coefficients of a sparse subset of the variables.

Keywords Disjunctive cuts · Epigraph · Sparsity · Valid inequalities

Mathematics Subject Classification 90C11 · 90C15

Parts of the paper have been published in proceedings of the 23rd International Conference on Integer Programming and Combinatorial Optimization, IPCO 2022. This research is supported by the Office of Naval Research under grant N00014-21-1-2574 and by NSF under grant 2000986.

✉ Rui Chen
rui.chen@cornell.edu

James Luedtke
jim.luedtke@wisc.edu

¹ Cornell Tech, New York, NY 10044, USA

² Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA

1 Introduction

We explore techniques for generating valid inequalities (cuts) for the epigraph E^Q of a function $Q : X \rightarrow \mathbb{R}$ over binary variables:

$$E^Q = \{(\theta, \mathbf{x}) \in \mathbb{R} \times X : \theta \geq Q(\mathbf{x})\}, \quad (1)$$

where $X \subseteq \{0, 1\}^n$. An important application motivating this study is stochastic mixed-integer programming (SMIP) [1], or more generally mixed-integer linear programs (MILPs) with block diagonal structures of the following form:

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} + \sum_{k=1}^N (\mathbf{d}^k)^T \mathbf{y}^k \\ \text{s.t.} \quad & T^k \mathbf{x} + W^k \mathbf{y}^k = \mathbf{h}^k, \mathbf{y}^k \geq \mathbf{0}, \quad k \in [N], \\ & \mathbf{x} \in X \subseteq \{0, 1\}^n. \end{aligned} \quad (2)$$

In the case of two-stage SMIPs, the binary variables \mathbf{x} represent first-stage decisions, N is the number of scenarios representing the possible outcomes, and for each $k \in [N] := \{1, \dots, N\}$, the continuous decision variables \mathbf{y}^k represent recourse actions taken in response to observing the data $(\mathbf{d}^k, T^k, W^k, \mathbf{h}^k)$ in scenario k . A common approach to solving such problems is Benders decomposition, which solves a reformulation of the form

$$\min_{\theta, \mathbf{x}} \left\{ \mathbf{c}^T \mathbf{x} + \sum_{k=1}^N \theta_k : \theta_k \geq Q_k(\mathbf{x}) \text{ for } k \in [N], \mathbf{x} \in X \right\}, \quad (3)$$

where for $k \in [N]$, $\mathbf{x} \in X$,

$$Q_k(\mathbf{x}) = \min_{\mathbf{y}} \{(\mathbf{d}^k)^T \mathbf{y} : T^k \mathbf{x} + W^k \mathbf{y} = \mathbf{h}^k, \mathbf{y} \geq \mathbf{0}\}. \quad (4)$$

The epigraph of Q_k of the form (1) shows up as a substructure in (3). In Benders decomposition, valid inequalities (Benders cuts [11, 54]) for this epigraph are derived via linear programming (LP) duality, but these are not generally sufficient to define the convex hull of the epigraph, thus motivating the need to derive stronger valid inequalities for sets of this form. This topic has been extensively studied both theoretically and computationally; see [17, 26, 30, 39, 47, 50, 52, 53, 56] as just a sample of the literature. Aside from SMIPs, this epigraph substructure appears in a variety of other problems (e.g., [12, 45, 55]).

We study a technique for generating inequalities for E^Q based on a disjunctive relaxation having many terms, specifically obtained by enumerating all $2^{|I|}$ feasible values for a subset I of the binary variables. Disjunctive programming has been a central tool in MILP since its origin in 1970s [4, 5]. A disjunction is a union of sets, and if the feasible region of an MILP is contained within such a union, inequalities valid for the disjunction are valid for the MILP, and are referred to as disjunctive cuts. Most disjunctive cuts used in practice are based on two disjunctive terms, e.g.,

split cuts [20] and lift-and-project cuts [6, 7, 9, 15]. While there has been significant work on classes of cuts that are derived from multiple-term disjunctions [2, 10, 22, 24, 42], the current methods remain focused on disjunctions with a relatively small number of terms. Perregaard and Balas [49] considered an iterative scheme for generating disjunctive cuts from many terms (see Sect. 2), but the approach remains computationally demanding. More recently, Balas and Kazachkov [8, 37] consider a \mathcal{V} -polyhedral perspective that allows generating and strengthening disjunctive cuts using up to 64 terms.

Our proposal for generating multi-term disjunctive cuts more efficiently is based on restricting the support of the generated cut to the index set I , the same set used to define the disjunctive terms. We refer to such cuts as I -sparse cuts. Our approach is motivated by the desire to generate sparse cuts, which may lead to faster solution time of the LP relaxations. Recent studies have investigated the theoretical strength of sparse cuts [23, 25, 26]. Our use of sparsity is with respect to the generated cut, which differentiates it from Fukasawa et al. [29] who empirically show that split cuts derived from (two-term) split disjunctions defined by a sparse integer vector can close the majority of the split closure gap. We also show that the idea of I -sparse cuts can be generalized to tilt a given valid inequality for a mixed-binary set, by modifying a particular subset of coefficients using a multi-term disjunction. Note that our use of the term “tilting” differs from that in [19, 27, 49], who use “tilting” to refer to a process of sequentially rotating a valid inequality with the goal of obtaining a facet-defining inequality. Our use of the term “tilting” instead refers to an approach for strengthening the coefficients of a subset of variables (while keeping the coefficients on the remaining variables fixed).

In Sect. 2 we show that the proposed sparsity restriction enables generating multi-term disjunctive cuts by solving a single subproblem per term, and then solving a single cut-generating LP. Thus, while this remains a computationally demanding cut generation process, we find empirically that it is feasible to use many more disjunctive terms than have previously been considered. In Sect. 3, we propose two rules for selecting the support I to generate I -sparse inequalities. In Sect. 4, we present results of a computational study using the I -sparse inequalities based on up to 2^{10} disjunctive terms on three test problems. We find that in many cases the I -sparse cuts close nearly as much gap as multi-term disjunctive cuts without the sparsity restriction, and can be generated orders of magnitude faster. When incorporated into a Benders branch-and-cut solution method, we find that I -sparse cuts lead to faster solution times or smaller ending gaps on our test instances. Although we find that sparse cuts often can close a significant portion of the optimality gap, we expect there are problems where dense cuts may be needed. Thus, we explore in Sect. 5 how we can use our proposed technique to optimally “tilt” a given (possibly dense) valid inequality by modifying a sparse subset of the coefficients of the inequality. We make concluding remarks in Sect. 6.

A preliminary version of this work appeared in the conference publication [18]. In this paper we include proofs of the main results, report new results from experiments dynamically adjusting the cardinality of the cut-support set and using the proposed cuts to solve problems to optimality within a cut-and-branch method, and derive the

approach for tilting a given valid inequality by modifying a sparse subset of its coefficients. Some notation has been changed to improve readability.

2 Sparse multi-term 0–1 disjunctive cuts

We study the problem of generating valid inequalities for the epigraph E^Q defined in (1). Let $R(X)$ be a (continuous) relaxation of X with $R(X) \cap \{0, 1\}^n = X$. We assume we have access to an extension of Q to $R(X)$, $q : R(X) \rightarrow \mathbb{R}$, satisfying $q(\mathbf{x}) = Q(\mathbf{x})$ for $\mathbf{x} \in X$. We require that minimizing q over $R(X)$ is efficiently solvable. For example, this would be the case if $R(X)$ is closed, convex, and equipped with an efficient separation oracle and q is convex over $R(X)$ with efficiently computable subgradients. We emphasize that we do not expect q to be the convex envelope of Q over X (i.e., $\text{conv}(E^Q)$) as our interest is precisely about identifying valid inequalities to approximate this set. In case an efficiently computable exact extension q is not readily available, one can use an extension q that instead satisfies $q(\mathbf{x}) \leq Q(\mathbf{x})$ for $\mathbf{x} \in X$. For example, in the case of an SMIP having integer second-stage decisions, the exact recourse function $Q_k(\mathbf{x})$ is nonconvex and expensive to evaluate, in which case one may use instead use the recourse function defined using an LP relaxation of the recourse problem. The strength of the resulting cuts will naturally depend on the quality of the relaxation, which could for example be improved using standard MILP valid inequalities.

The following example provides another illustration of the choice of q .

Example 1 ([41]) Assume $Q : \{0, 1\}^n \rightarrow \mathbb{R}$ is defined by

$$Q(\mathbf{x}) = \frac{\mathbf{a}^T \mathbf{x} + b}{\mathbf{c}^T \mathbf{x} + d}$$

with $\mathbf{a} \in \mathbb{R}_+^n$, $\mathbf{c} \in \mathbb{R}_+^n$ and $d > 0$. The natural continuous extension of Q to $[0, 1]^n$ is not necessarily convex. However, it is possible [41] to construct a convex extension of Q over $R(X) = [0, 1]^n$ by introducing a new variable $y = 1/(\mathbf{c}^T \mathbf{x} + d) \geq 0$ and linearization variables $z_i = x_i y \geq 0$ for $i \in [n]$, and define the function $q : [0, 1]^n \rightarrow \mathbb{R}$ by

$$q(\mathbf{x}) = \min_{y, \mathbf{z}} \mathbf{a}^T \mathbf{z} + by \tag{5a}$$

$$\text{s.t. } z_i - y \leq 0, \quad (c_i + d)z_i \leq x_i, \quad d(y - z_i) \leq 1 - x_i, \quad i \in [n], \tag{5b}$$

$$\mathbf{c}^T \mathbf{z} + dy = 1, \quad \mathbf{z} \in \mathbb{R}_+^n, \quad y \geq 0 \tag{5c}$$

for $\mathbf{x} \in [0, 1]^n$. Then, q is convex over $[0, 1]^n$ (because it is the value function of a linear program with \mathbf{x} in the right-hand side of the constraints) and $q(\mathbf{x}) = Q(\mathbf{x})$ for $\mathbf{x} \in \{0, 1\}^n$, and thus it is a convex extension of Q . Finally, observe that evaluating q and obtaining a subgradient of q at a point \mathbf{x} can be accomplished by solving the linear program (5).

Let $E^q := \{(\theta, \mathbf{x}) \in \mathbb{R} \times R(X) : \theta \geq q(\mathbf{x})\}$ denote the epigraph of q over $R(X)$ and let I be a nonempty subset of $[n]$. We denote by \mathbf{x}_I the subvector of \mathbf{x} with indices I , and define $\{0, 1\}^I := \{\mathbf{x}_I : x_i \in \{0, 1\}, i \in I\}$. For each given $\chi \in \{0, 1\}^I$, we define

$$E_I^q(\chi) := \{(\theta, \mathbf{x}) \in E^q : \mathbf{x}_I = \chi\} = \{(\theta, \mathbf{x}) \in \mathbb{R} \times R(X) : \theta \geq q(\mathbf{x}), \mathbf{x}_I = \chi\}.$$

We derive valid inequalities for E^Q by finding valid inequalities for the following multi-term disjunctive relaxation of E^Q :

$$E_I^q := \bigcup_{\chi \in \{0, 1\}^I} E_I^q(\chi). \quad (6)$$

Since $E^Q \subseteq E_I^q$, any inequality valid for E_I^q is also valid for E^Q . We call the relaxation E_I^q of E^Q a multi-term 0-1 disjunction, and any cut valid for E_I^q a multi-term 0-1 disjunctive cut. We include q as a superscript in the notation $E_I^q(\chi)$ and E_I^q to emphasize that these relaxations depend on the choice of the extension q . These relaxations also depend on the choice of $R(X)$, i.e., the domain of q , but we suppress this dependence for notational convenience.

2.1 Generating multi-term 0–1 disjunctive cuts

By (6), an inequality of the form $\mu_0\theta - \mu^T \mathbf{x} \geq \eta$ is valid for E_I^q if and only if

$$\min_{\theta, \mathbf{x}} \left\{ \mu_0\theta - \mu^T \mathbf{x} : (\theta, \mathbf{x}) \in E_I^q(\chi) \right\} \geq \eta \text{ for all } \chi \in \{0, 1\}^I. \quad (7)$$

Therefore, to separate a point $(\hat{\theta}, \hat{\mathbf{x}})$ from E_I^q , in principle one can solve the following problem:

$$\min_{\mu_0, \mu, \eta} \mu_0\hat{\theta} - \mu^T \hat{\mathbf{x}} - \eta \quad (8a)$$

$$\text{s.t. } \mu_0\theta - \mu^T \mathbf{x} \geq \eta, \forall (\mathbf{x}, \theta) \in E_I^q(\chi), \chi \in \{0, 1\}^I, \quad (8b)$$

$$\mu_0 \geq 0, \|(\mu_0, \mu)\|_1 \leq 1, \quad (8c)$$

where (8c) is just one example of a normalization constraint that can be used to ensure the separation problem has an optimal solution.

Perregaard and Balas [49] suggest an iterative row generating algorithm for generating multi-term disjunctive cuts. Adapting it to our multi-term 0-1 disjunction leads to Algorithm 1 for solving (8). Specifically, the method alternates between solving a relaxation of (8) defined by only including constraints (8b) for a (small) subset of the extreme points of $E_I^q(\chi)$ for each $\chi \in \{0, 1\}^I$ (line 5), and then solving a subproblem for each $\chi \in \{0, 1\}^I$ to determine if any of the excluded constraints in (8b) is violated (line 7) and adding one such constraint if so. While this approach is guaranteed to yield a valid inequality for E_I^q that cuts off $(\hat{\theta}, \hat{\mathbf{x}})$ when one exists, the scalability of

Algorithm 1: The row generating algorithm for solving (8)

```

1 Input:  $I \subseteq [n]$ 
2 Output: A valid inequality  $\hat{\mu}_0 \theta - \hat{\mu}^T \mathbf{x} \geq \hat{\eta}$  for  $E_I^q$ 
3 Initialize a set  $\hat{S}_\chi$  as a subset of extreme points of  $E_I^q(\chi)$  for each  $\chi \in \{0, 1\}^I$ ;
4 repeat
5   Compute an optimal solution  $(\hat{\mu}_0, \hat{\mu}, \hat{\eta})$  of the LP:
      
$$\begin{aligned} \min_{\mu_0, \mu, \eta} \quad & \mu_0 \hat{\theta} - \mu^T \hat{\mathbf{x}} - \eta \\ \text{s.t.} \quad & \mu_0 \theta - \mu^T \mathbf{x} \geq \eta, \quad \forall (\theta, \mathbf{x}) \in \hat{S}_\chi, \quad \chi \in \{0, 1\}^I, \\ & \mu_0 \geq 0, \|\mu_0, \mu\|_1 \leq 1; \end{aligned}$$

6   for  $\chi \in \{0, 1\}^I$  do
7     Solve
      
$$\eta(\hat{\mu}_0, \hat{\mu}; \chi) := \min_{(\theta, \mathbf{x}) \in E_I^q(\chi)} \hat{\mu}_0 \theta - \hat{\mu}^T \mathbf{x}; \tag{9}$$

8     if  $\eta(\hat{\mu}_0, \hat{\mu}; \chi) < \hat{\eta}$  then
9       | Add an optimal solution  $(\theta^*, \mathbf{x}^*)$  of (9) into  $\hat{S}_\chi$ ;
10    end
11  end
12 until  $\eta(\hat{\mu}_0, \hat{\mu}; \chi) \geq \hat{\eta}$  for all  $\chi \in \{0, 1\}^I$ ;

```

the algorithm is limited by the multiplied effect of (a) the size of $\{0, 1\}^I$, and (b) the potential need to solve (9) multiple times for each $\chi \in \{0, 1\}^I$. Numerical experiments in [49] generate valid inequalities for MILPs using only up to 16 disjunctive terms. Balas and Kazachkov [8, 37] explore an approach for expanding the number of terms that can be included that is based on using a well-chosen relaxation of each term, which enables generating and strengthening disjunctive cuts using up to 64 terms. In this work, we propose a complementary approach in which we restrict attention to cuts supported on I , which we find eliminates the effect of (b), and enables us to conduct experiments with up to 1024 terms.

2.2 I -sparse inequalities

We next explore how restricting the support of the generated cut can be used to accelerate the generation of multi-term 0-1 disjunction cuts for E_I^q for a fixed I .

Definition 1 Let $I \subseteq [n]$. We say an inequality $\theta \geq \mu^T \mathbf{x} + \eta$ is an I -sparse inequality/cut for E^Q (with respect to the extension q) if the following two conditions hold:

1. $\theta \geq \mu^T \mathbf{x} + \eta$ is valid for E_I^q ;
2. $\mu_i = 0$ for all $i \notin I$.

The following proposition characterizes I -sparse inequalities.

Proposition 1 *An inequality $\theta \geq \mu^T \mathbf{x} + \eta$ with $\mu_i = 0$ for all $i \notin I$ is an I -sparse inequality for E^Q if and only if*

$$\mu_I^T \chi + \eta \leq v_I^q(\chi), \quad \forall \chi \in \{0, 1\}^I, \quad (10)$$

where for each $\chi \in \{0, 1\}^I$,

$$v_I^q(\chi) := \min\{q(\mathbf{x}) : \mathbf{x} \in R(X), \mathbf{x}_I = \chi\}. \quad (11)$$

Proof We only need to show that (7) with $\mu_0 = 1$ and $\mu_{[n] \setminus I} = \mathbf{0}$ holds if and only if (10) holds. This is straightforward by observing that for each $\chi \in \{0, 1\}^I$,

$$\begin{aligned} \min_{\theta, \mathbf{x}} \left\{ \theta - \sum_{i \in I} \mu_i x_i : (\theta, \mathbf{x}) \in E_I^q(\chi) \right\} \\ &= \min_{\theta, \mathbf{x}} \left\{ \theta - \mu_I^T \chi : (\theta, \mathbf{x}) \in E_I^q(\chi) \right\} \\ &= \min_{\theta, \mathbf{x}} \left\{ \theta : (\theta, \mathbf{x}) \in E_I^q(\chi) \right\} - \mu_I^T \chi \\ &= \min_{\theta, \mathbf{x}} \left\{ \theta : \theta \geq q(\mathbf{x}), \mathbf{x} \in R(X), x_I = \chi \right\} - \mu_I^T \chi \\ &= v_I^q(\chi) - \mu_I^T \chi. \end{aligned}$$

□

Observe that the problem (11) has a similar form as (9) which is used when applying the Perregaard and Balas algorithm [49] to solve (8).

Remark 1 Our presentation focuses on the case where $X \subseteq \{0, 1\}^n$. However, the result naturally extends to problems with bounded integer variables by enumerating all possible combinations of values that a subset of the bounded integer variables can take.

The following result provides a condition under which every nontrivial valid inequality for E^Q with coefficients supported on the index set I is an I -sparse inequality.

Corollary 1 *If $X = \{0, 1\}^n$, $R(X) = [0, 1]^n$ and q is component-wise monotonically nonincreasing or nondecreasing on $R(X)$, then an inequality $\theta \geq \mu^T \mathbf{x} + \eta$ with $\mu_i = 0$ for all $i \notin I$ is valid for E^Q if and only if it is an I -sparse inequality.*

Proof As in the proof of Proposition 1, an inequality $\theta \geq \mu^T \mathbf{x} + \eta$ with $\mu_{[n] \setminus I} = \mathbf{0}$ is valid for E^Q if and only if

$$\min\{q(\mathbf{x}) : \mathbf{x}_I = \chi, \mathbf{x} \in \{0, 1\}^n\} \geq \sum_{i \in I} \mu_i \chi_i + \eta, \quad \forall \chi \in \{0, 1\}^I.$$

The conclusion then follows by observing that, for each $\chi \in \{0, 1\}^I$,

$$v_I^q(\chi) = \min\{q(\mathbf{x}) : \mathbf{x}_I = \chi, \mathbf{x} \in [0, 1]^n\}$$

$$= \min\{q(\mathbf{x}) : \mathbf{x}_I = \chi, \mathbf{x} \in \{0, 1\}^n\},$$

where the last equality is due to the fact that a monotone function over a box always attains its minimum at an extreme point. \square

When the assumptions of Corollary 1 do not hold, Proposition 1 still provides a method for separating I -sparse cuts – we just cannot assure in this case that I -sparse cuts contain all cuts supported only on I .

Based on Proposition 1, for a fixed I , the separation problem for I -sparse inequalities for a point $(\hat{\theta}, \hat{\mathbf{x}})$ can be solved by solving the LP

$$g_{\hat{\mathbf{x}}}^q(I) = \max \left\{ \sum_{i \in I} \mu_i \hat{x}_i + \eta : \mu_I^T \chi + \eta \leq v_I^q(\chi), \chi \in \{0, 1\}^I \right\}. \quad (12)$$

Specifically, the optimal solution of (12) defines an inequality that cuts off $(\hat{\theta}, \hat{\mathbf{x}})$ if and only if $g_{\hat{\mathbf{x}}}^q(I) > \hat{\theta}$. Since q is finite valued in $R(X)$, $v_I^q(\chi) \in \mathbb{R}$ for $\chi \in \text{proj}_I(R(X)) \cap \{0, 1\}^I$. When $\hat{\mathbf{x}}_I \in \text{conv}(\text{proj}_I(R(X)) \cap \{0, 1\}^I)$, the LP (12) is guaranteed to have an optimal solution. When $\hat{\mathbf{x}}_I \notin \text{conv}(\text{proj}_I(R(X)) \cap \{0, 1\}^I)$, $(\hat{\theta}, \hat{\mathbf{x}})$ can be cut off by an inequality separating $\hat{\mathbf{x}}_I$ from $\text{proj}_I(R(X)) \cap \{0, 1\}^I$, which can be generated by enumerating $\text{proj}_I(R(X)) \cap \{0, 1\}^I$.

The main work to generate an I -sparse inequality is evaluating $v_I^q(\chi)$ by solving (11) for each $\chi \in \{0, 1\}^I$, and then solving the LP (12) once. Note that (12) has $|I| + 1$ variables in contrast to $n + 2$ variables in the problem (8) used in the Perregaard and Balas (PB) [49] algorithm, and requires solving at most $2^{|I|}$ subproblems of the form (11), in contrast to the PB algorithm which solves $2^{|I|}$ subproblems of this form in multiple iterations until convergence.

2.3 Accelerating cut generation

Evaluating $v_I^q(\chi)$ for all $\chi \in \{0, 1\}^I$ is the most significant computational component of generating an I -sparse inequality. We discuss techniques to potentially accelerate this evaluation, focusing on our motivating example of MILPs with block diagonal structures (2). In this context, assume $R(X) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$ is a polyhedral relaxation of X and assume bound constraints $0 \leq x_i \leq 1$ are included in $\mathbf{Ax} \leq \mathbf{b}$. For a fixed $k \in [N]$, let $Q_k(\mathbf{x})$ be as defined in (4) and assume $Q_k(\mathbf{x})$ is finite valued for all $\mathbf{x} \in R(X)$. In this case, when generating an I -sparse inequality for the set $E^{Q_k} = \{(\theta_k, \mathbf{x}) \in \mathbb{R} \times X : \theta_k \geq Q_k(\mathbf{x})\}$ the evaluation of $v_I^q(\chi)$ for $\chi \in \{0, 1\}^I$ can be formulated as the following LP

$$v_I^q(\chi) = \min_{\mathbf{x}, \mathbf{y}} \{(\mathbf{d}^k)^T \mathbf{y}^k : T^k \mathbf{x} + W^k \mathbf{y} = \mathbf{h}^k, \mathbf{y} \geq \mathbf{0}, \mathbf{Ax} \leq \mathbf{b}, \mathbf{x}_I = \chi\}. \quad (13)$$

A first simple idea for accelerating the solution of (13) for all $\chi \in \{0, 1\}^I$ is to exploit the possibility to warm-start these LPs (see, e.g., [13] for background). LP solvers like Gurobi [36] automatically implement a simplex warm start when only variable bounds

are changed in an LP. Thus, solving the sequence of problems (13) for $\chi \in \{0, 1\}^I$ by making changes to variable bounds implied by the constraints $\mathbf{x}_I = \chi$ will naturally benefit from these warm-start capabilities. This motivates a careful selection of the sequence these problems are solved in. For example, by following the sequence defined by a Gray code [31], at most one variable bound will change from one subproblem to the next.

We do not explore this in our computational study, but another possibility for reducing the time required for evaluating $v_I^q(\chi)$ is to use a lower bound on q that is simpler to evaluate. For example, for MILPs with block diagonal structure, a lower bound on $v_I^{Q_k}(\chi)$ is obtained by solving a problem of the form:

$$v_I^{\hat{Q}_k}(\chi) = \min\{\hat{Q}_k(\mathbf{x}) : \mathbf{x} \in R(X), \mathbf{x}_I = \chi\}$$

where \hat{Q}_k is the current piecewise-linear convex lower bound of Q_k defined by Benders cuts. These lower bounds could then be used in (12) which would yield a valid but potentially weaker inequality. This inequality could then be improved by exactly evaluating $v_I^{Q_k}(\chi)$ for the χ that correspond to binding constraints in (12), and then re-solving (12) with these improved values.

Finally, we note that after evaluating $v_I^q(\chi)$ for $\chi \in \{0, 1\}^I$ for a given set I (and potentially adding a cut based on solving (12)), we recommend storing these values for future use. In particular, after re-solving the LP relaxation after addition of cuts and obtaining a new candidate relaxation solution $(\hat{\theta}, \hat{\mathbf{x}})$, it may be possible that solving (12) again for the same set I can lead to a new violated inequality. Storing the values $v_I^q(\chi)$ for $\chi \in \{0, 1\}^I$ avoids needing to re-calculate them, so that only (12) needs to be solved to determine if such a violated inequality exists.

3 Two selection rules for the support I

We now discuss techniques for choosing the set I when generating I -sparse cuts. Given a point $(\hat{\theta}, \hat{\mathbf{x}})$, the goal is to select I in order to maximize the cut violation $g_{\hat{\mathbf{x}}}^q(I)$ (defined in (12)). Since the complexity of generating these cuts grows exponentially with $|I|$ we investigate techniques that choose I satisfying $|I| \leq K$ for some fixed (small) integer K . We describe two selection rules that are derived from two different approximations of q .

3.1 A greedy rule based on a monotone submodular approximation

The problem of choosing I that maximizes $g_{\hat{\mathbf{x}}}^q(I)$ is a set function optimization problem. For notational convenience, we do not distinguish between a set function and a function with binary variables, i.e., we interchangeably use $f(A)$ for $f(\chi_A)$ for all $A \subseteq [n]$ where $\chi_A \in \{0, 1\}^n$ is the indicator vector of A . One particular class of set functions satisfying good theoretical properties is monotone submodular functions [33].

Definition 2 A function $f : 2^{[n]} \rightarrow \mathbb{R}$ is monotone submodular if it satisfies the following two conditions:

1. (Monotonicity) If $S \subseteq T \subseteq [n]$, then $f(S) \leq f(T)$;
2. (Submodularity) If $S \subseteq [n]$, $j, k \in [n] \setminus S$ and $j \neq k$, then $f(S \cup \{j\}) + f(S \cup \{k\}) \geq f(S \cup \{j, k\}) + f(S)$.

Given $\hat{\mathbf{x}} \in [0, 1]^n$, we can show that the cut violation function $g_{\hat{\mathbf{x}}}^q(I)$ is monotone submodular in I if Q is monotone submodular and its extension q is component-wise monotonically nondecreasing.

Proposition 2 Assume $X = \{0, 1\}^n$, $R(X) = [0, 1]^n$, Q is monotone submodular on X , and its extension q is component-wise monotonically nondecreasing on $R(X)$. Then the cut violation function $g_{\hat{\mathbf{x}}}^q$ is monotone submodular.

Proof The monotonicity of $g_{\hat{\mathbf{x}}}^q$ is obvious since an I -sparse inequality is also I' -sparse for any $I \subseteq I' \subseteq [n]$. We only need to show submodularity of $g_{\hat{\mathbf{x}}}^q$.

For all $I \subseteq [n]$, let $Q_I : \{0, 1\}^I \rightarrow \mathbb{R}$ be the function with $Q_I(\chi_S) = Q(S) = Q(\chi_S)$ for all $S \subseteq I$ and let $\bar{Q}_I : [0, 1]^I \rightarrow \mathbb{R}$ denote the convex envelope of Q_I on $[0, 1]^I$. By component-wise monotonicity of q on $R(X)$,

$$v_I^q(\chi) = \min\{Q(\mathbf{x}) : \mathbf{x}_I = \chi, \mathbf{x} \in \{0, 1\}^n\} = Q_I(\chi), \quad \forall \chi \in \{0, 1\}^I.$$

Therefore,

$$\begin{aligned} g_{\hat{\mathbf{x}}}^q(I) &= \max_{\mu, \eta} \sum_{i \in I} \mu_i \hat{x}_i + \eta, \\ \text{s.t. } \mu_I^T \chi + \eta &\leq Q_I(\chi), \quad \forall \chi \in \{0, 1\}^I. \end{aligned} \quad (14)$$

Then by [51, Corollary 12.1.1], $g_{\hat{\mathbf{x}}}^q(I) = \bar{Q}_I(\hat{\mathbf{x}}_I)$ as (14) characterizes all affine underestimates of Q_I . Note that Q_I is submodular by submodularity of Q . Then the convex envelope \bar{Q}_I is characterized by the Lovász extension [44] of Q_I . We are now ready to show that $g_{\hat{\mathbf{x}}}^q$ is submodular. Let $S \subseteq [n]$ and $j, k \in [n] \setminus S$ such that $j \neq k$. Let i_1, \dots, i_m be a reordering of elements in $S \cup \{j, k\}$ such that $\hat{x}_{i_1} \geq \dots \geq \hat{x}_{i_m}$. Define $S_l := \{i_1, \dots, i_l\}$ for $l = 1, \dots, m$. Using Lovász extensions of Q_S , $Q_{S \cup \{j\}}$, $Q_{S \cup \{k\}}$ and $Q_{S \cup \{j, k\}}$, we have

1. $g_{\hat{\mathbf{x}}}^q(S \cup \{j, k\}) = (1 - \hat{x}_{i_1})Q(\emptyset) + \sum_{l=1}^{m-1} (\hat{x}_{i_l} - \hat{x}_{i_{l+1}})Q(S_l) + \hat{x}_{i_m}Q(S_{i_m})$;
2. $g_{\hat{\mathbf{x}}}^q(S \cup \{j\}) = (1 - \hat{x}_{i_1})Q(\emptyset) + \sum_{l=1}^{m-1} (\hat{x}_{i_l} - \hat{x}_{i_{l+1}})Q(S_l \setminus \{k\}) + \hat{x}_{i_m}Q(S_{i_m} \setminus \{k\})$;
3. $g_{\hat{\mathbf{x}}}^q(S \cup \{k\}) = (1 - \hat{x}_{i_1})Q(\emptyset) + \sum_{l=1}^{m-1} (\hat{x}_{i_l} - \hat{x}_{i_{l+1}})Q(S_l \setminus \{j\}) + \hat{x}_{i_m}Q(S_{i_m} \setminus \{j\})$;
4. $g_{\hat{\mathbf{x}}}^q(S) = (1 - \hat{x}_{i_1})Q(\emptyset) + \sum_{l=1}^{m-1} (\hat{x}_{i_l} - \hat{x}_{i_{l+1}})Q(S_l \setminus \{j, k\}) + \hat{x}_{i_m}Q(S_{i_m} \setminus \{j, k\})$.

Note that $Q(S_l \setminus \{k\}) + Q(S_l \setminus \{j\}) \geq Q(S_l) + Q(S_l \setminus \{j, k\})$ for $l = 1, \dots, m$ due to submodularity and monotonicity of Q . It follows that $g_{\hat{\mathbf{x}}}^q(S \cup \{j\}) + g_{\hat{\mathbf{x}}}^q(S \cup \{k\}) \geq g_{\hat{\mathbf{x}}}^q(S \cup \{j, k\}) + g_{\hat{\mathbf{x}}}^q(S)$, i.e., $g_{\hat{\mathbf{x}}}^q$ is submodular. \square

Although maximizing a monotone submodular function subject to a cardinality constraint is NP-hard [21] in general, the well-known greedy algorithm of Nemhauser

Algorithm 2: Greedy algorithm for choosing I

```

1 Input:  $\hat{\mathbf{x}}, K$ 
2 Output:  $I$ 
3 Initialize  $I \leftarrow \emptyset$ 
4 while  $|I| \leq K$  do
5   Evaluate  $g_{\hat{\mathbf{x}}}^q(I \cup \{i\})$  for each  $i \notin I$ ;
6    $I \leftarrow I \cup \{i^*\}$  where  $i^* \in \arg \max_{i \notin I} g_{\hat{\mathbf{x}}}^q(I \cup \{i\})$ ;
7 end

```

et al. [46] attains a $1 - 1/e$ approximation ratio to this problem. For maximizing $g_{\hat{\mathbf{x}}}^q(I)$ subject to a cardinality constraint $|I| \leq K$, the greedy algorithm is described in Algorithm 2. However, directly applying a greedy algorithm for choosing I may not be a good choice because (i) the assumptions of Proposition 2 may not hold, and (ii) the greedy algorithm requires evaluating $g_{\hat{\mathbf{x}}}^q$ many times, which is computationally expensive. Therefore, we seek alternatives to this approach by applying the greedy algorithm to a different cut violation function $g_{\hat{\mathbf{x}}}^{\tilde{Q}}$ associated with function $\tilde{Q} : [0, 1]^n \rightarrow \mathbb{R}$, whose restriction \tilde{Q}' on $\{0, 1\}^I$ is an approximation of the function Q . We choose \tilde{Q} such that \tilde{Q}' is monotone and submodular and the cut violation $g_{\hat{\mathbf{x}}}^{\tilde{Q}}$ can be evaluated much more efficiently than $g_{\hat{\mathbf{x}}}^q$.

We propose to use \tilde{Q} of the form $\tilde{Q}(\mathbf{x}) = \max_{i \in [n]} \{a_i x_i + b\}$ with $0 \leq a_1 \leq \dots \leq a_n$ (after complementing and reordering some variables). With this form, \tilde{Q} is component-wise nondecreasing on $R(X)$ and \tilde{Q}' (the restriction of \tilde{Q} to X) is monotone submodular on X , and thus the associated approximation $g_{\hat{\mathbf{x}}}^{\tilde{Q}}$ is monotone submodular. To construct such an approximation \tilde{Q} , we use the I -sparse inequalities with $I = \{i\}$ for each $i \in [n]$. When $I = \{i\}$, the polyhedron defined by (10) has a unique extreme point $(v_{\{i\}}^q(1) - v_{\{i\}}^q(0), v_{\{i\}}^q(0))$, which corresponds to a valid inequality of E^Q :

$$\theta \geq (v_{\{i\}}^q(1) - v_{\{i\}}^q(0))x_i + v_{\{i\}}^q(0). \quad (15)$$

By complementing the variable $x_i \leftarrow 1 - x_i$ if necessary, we may assume that $v_{\{i\}}^q(1) \geq v_{\{i\}}^q(0)$. Thus, $Q(\mathbf{x}) \geq \text{LB}^* := \max_{i \in [n]} v_{\{i\}}^q(0)$ for all $\mathbf{x} \in \{0, 1\}^n$. Therefore, we can strengthen (15) to be $\theta \geq (\tilde{v}_{\{i\}}^q(1) - \text{LB}^*)x_i + \text{LB}^*$, where $\tilde{v}_{\{i\}}^q(1) = \max\{v_{\{i\}}^q(1), \text{LB}^*\}$ for all $i \in [n]$. We thus obtain inequalities of the form $\theta \geq a_i x_i + b$ for $i \in [n]$ with $a_i \geq 0$, which are valid for E^Q (modulo the mentioned complementing of the x_i variables as needed). Assuming without loss of generality that $a_1 \leq \dots \leq a_n$, we obtain the desired approximation \tilde{Q} .

We next discuss how to use $g_{\hat{\mathbf{x}}}^{\tilde{Q}}$ to generate a support within Algorithm 2. In particular, we discuss how to efficiently evaluate $g_{\hat{\mathbf{x}}}^{\tilde{Q}}(I)$ for a subset I . Define $\tilde{Q}_I : \{0, 1\}^I \rightarrow \mathbb{R}$ by

$$\tilde{Q}_I(\mathbf{x}) = \max_{i \in I} \{a_i x_i + b\} \quad \text{for all } \mathbf{x} \in \{0, 1\}^I$$

and let \bar{Q}_I be the convex envelope of \tilde{Q}_I over $[0, 1]^I$. As discussed in the proof of Proposition 2, submodularity of $g_{\hat{\mathbf{x}}}^{\tilde{Q}}$ implies that $g_{\hat{\mathbf{x}}}^{\bar{Q}}(I) = \bar{Q}_I(\hat{\mathbf{x}}_I)$. The convex envelope of \tilde{Q}_I over $[0, 1]^I$ is the convex hull of the set:

$$F_I = \{(\theta, \mathbf{x}) \in \mathbb{R} \times \{0, 1\}^I : \theta \geq a_i x_i + b, i \in I\}. \quad (16)$$

The convex hull of this set has been characterized in [3, 35].

Theorem 1 ([3, 35]) *Assume $I = \{1, 2, \dots, d\}$ with $0 \leq a_1 \leq \dots \leq a_d$. Then*

$$\begin{aligned} \text{conv}(F_I) = \{(\theta, \mathbf{x}) \in \mathbb{R} \times [0, 1]^d : \theta \geq a_{i_1} x_{i_1} + \sum_{k=2}^m (a_{i_k} - a_{i_{k-1}}) x_{i_k} + b, \\ \text{for all subsequences } (i_k)_{k=1}^m \text{ of } [d] \text{ such that } 1 \leq i_1 \leq \dots \leq i_m = d\}. \end{aligned} \quad (17)$$

Thus, for a given $\hat{\mathbf{x}}$, the problem of evaluating $\bar{Q}_I(\hat{\mathbf{x}}_I)$ can be posed as $\min\{\theta : (\theta, \hat{\mathbf{x}}_I) \in \text{conv}(F_I)\}$ which is equivalent to finding the inequality in the family of inequalities given in (17) with maximum right-hand side when evaluated at $\hat{\mathbf{x}}_I$. This, in turn, is equivalent to the separation problem of this class of inequalities, which can be solved in polynomial time [3, 35]. We describe the application of the separation algorithm from [35] to this context in Algorithm 3. Incorporating this approach for evaluating $g_{\hat{\mathbf{x}}}^{\bar{Q}}(I)$ into the greedy algorithm yields a much quicker method for choosing I than using the greedy algorithm with exact evaluation of $g_{\hat{\mathbf{x}}}^Q(I)$. Indeed, the most significant work in this case is solving the problem (11) with $\chi = 0$ and $\chi = 1$ for each $i \in [n]$ to obtain the values $v_{[i]}^q(1)$ and $v_{[i]}^q(0)$ for $i \in [n]$, which only needs to be done once for the overall greedy algorithm.

The proposed approximation may not lead to a good choice of I when \tilde{Q} is not a good approximation of Q , in particular because the approximation \tilde{Q} is based on the maximum of affine lower bounding functions, each supported by a single variable. Thus, it is natural to consider using affine functions with more general support to build the lower bounding approximation. However, unless $P = NP$, the following result and the equivalence between optimization and separation [32] indicate that the key step of evaluating the convex envelope of the given function would no longer be efficiently solvable even if the support of the inequalities defining the lower approximation of Q were restricted to just two variables per inequality.

Proposition 3 *It is NP-hard to optimize a linear function over*

$$\{(\theta, \mathbf{x}) \in \mathbb{R} \times \{0, 1\}^n : \theta \geq \mathbf{a}^T \mathbf{x} + b, (\mathbf{a}, b) \in \mathcal{A}\} \quad (18)$$

even if $\|\mathbf{a}\|_0 \leq 2$ for each $(\mathbf{a}, b) \in \mathcal{A}$ and $|\mathcal{A}|$ is polynomially bounded by n .

Proof We prove by polynomially reducing an arbitrary instance of the NP-complete vertex cover problem to a linear optimization problem over (18) with $\|\mathbf{a}\|_0 \leq 2$ and $|\mathcal{A}| = O(n^2)$. The vertex cover problem is stated as:

Algorithm 3: Evaluating the violation underestimate $g_{\hat{\mathbf{x}}}^{\tilde{Q}}(I)$.

```

1 Input:  $\hat{\mathbf{x}}, I = \{1, \dots, d\}, b, a_i, i \in I$  with  $0 \leq a_1 \leq \dots \leq a_d$ 
2 Output:  $g_{\hat{\mathbf{x}}}^{\tilde{Q}}(I)$ 
3 Initialize  $\hat{x}_{\max} \leftarrow -\infty, i_{\max} \leftarrow d, (\sigma_i)_{i=1}^d \leftarrow (\emptyset)_{i=1}^d$ ;
4 for  $i = d, d-1, \dots, 1$  do
5   if  $\hat{x}_i > \hat{x}_{\max}$  then
6      $\sigma_i \leftarrow i_{\max}, \hat{x}_{\max} \leftarrow \hat{x}_i, i_{\max} \leftarrow i$ ;
7   end
8 end
9  $\tilde{g} \leftarrow b + a_{i_{\max}} \hat{x}_{i_{\max}}, k \leftarrow i_{\max}$ ;
10 while  $k \neq d$  do
11    $\tilde{g} \leftarrow \tilde{g} + (a_{\sigma_k} - a_k) \hat{x}_{\sigma_k}$ ;
12    $k \leftarrow \sigma_k$ ;
13 end
14 Return  $g_{\hat{\mathbf{x}}}^{\tilde{Q}}(I) = \tilde{g}$ ;

```

- Given an undirected graph $G = (V, E)$ and positive integer k' , does there exist $V' \subseteq V$ with $|V'| \leq k'$ such that $u \in V'$ or $v \in V'$ for each $uv \in E$?

We next show that such vertex cover V' exists if and only if the optimal objective value of the following problem is at most -1 :

$$\min \left\{ \sum_{v \in V} x_v + (k+1)\theta : \theta \geq -1; \theta \geq -x_u - x_v, uv \in E; \mathbf{x} \in \{0, 1\}^V \right\}. \quad (19)$$

Note that (19) ≤ -1 if and only if the optimal solution (θ^*, \mathbf{x}^*) satisfies $\theta^* = -1$, $\sum_{v \in V} x_v^* \leq k$ and $-1 \geq -x_u^* - x_v^*$ for each $uv \in E$. Such \mathbf{x}^* corresponds to a vertex cover $V' := \{v \in V : x_v^* = 1\}$ of G with $|V'| \leq k'$. On the other hand, a vertex cover V' of G with $|V'| \leq k'$ corresponds to an optimal solution $(-1, \mathbf{x}^*)$ of (19) with objective value at most -1 satisfying $x_v^* = 1$ if and only if $v \in V'$. \square

3.2 A cutting-plane approximation rule

We next describe an alternative selection rule for I that is based on a single affine lower bound (e.g., from a cutting-plane) of Q . Let $\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}$, and

$$F_{(\mathbf{a}, b)} = \{(\theta, \mathbf{x}) \in \mathbb{R} \times \{0, 1\}^n : \theta \geq \mathbf{a}^T \mathbf{x} + b\}.$$

Let $(\hat{\theta}, \hat{\mathbf{x}}) \in \mathbb{R} \times [0, 1]^n$ be given, and consider the problem of finding a valid inequality for $F_{(\mathbf{a}, b)}$ of the form

$$\theta \geq \sum_{i \in I} \mu_i x_i + \eta \quad (20)$$

that is maximally violated by $(\hat{\theta}, \hat{\mathbf{x}})$.

Proposition 4 *The problem of maximizing*

$$\sum_{i \in I} \mu_i \hat{x}_i + \eta - \hat{\theta}$$

such that inequality (20) defined by μ, η is valid for $F_{(\mathbf{a}, b)}$ has optimal value $-\sum_{i=1}^n a_i^- + \sum_{i \in I} (a_i \hat{x}_i + a_i^-) + b - \hat{\theta}$, where $a_i^- = \max\{-a_i, 0\}$.

Proof Inequality $\theta \geq \sum_{i \in I} \mu_i x_i + \eta$ is valid for $F_{(\mathbf{a}, b)}$ if and only if

$$\begin{aligned} \mu_I^T \chi + \eta &\leq \min\{\theta : (\theta, \mathbf{x}) \in F_{(\mathbf{a}, b)}, \mathbf{x}_I = \chi\} \\ &= \sum_{i \in I} a_i \chi_i - \sum_{i \notin I} a_i^- + b, \quad \forall \chi \in \{0, 1\}^I. \end{aligned}$$

Therefore, by LP duality, the maximum violation of an inequality of this form is

$$\begin{aligned} &\max_{\mu, \eta} \left\{ \sum_{i \in I} \mu_i \hat{x}_i + \eta - \hat{\theta} : \mu_I^T \chi + \eta \leq \sum_{i \in I} a_i \chi_i - \sum_{i \notin I} a_i^- + b, \chi \in \{0, 1\}^I \right\} \\ &= \min \left\{ \sum_{\chi \in \{0, 1\}^I} \sum_{i \in I} a_i \chi_i \lambda_\chi + \sum_{\chi \in \{0, 1\}^I} \left(- \sum_{i \notin I} a_i^- + b \right) \lambda_\chi : \right. \\ &\quad \left. \sum_{\chi \in \{0, 1\}^I} \chi_i \lambda_\chi = \hat{x}_i, i \in I; \sum_{\chi \in \{0, 1\}^I} \lambda_\chi = 1, \lambda \geq 0 \right\} - \hat{\theta} \\ &= \sum_{i \in I} a_i \hat{x}_i - \sum_{i \notin I} a_i^- + b - \hat{\theta} \\ &= - \sum_{i=1}^n a_i^- + \sum_{i \in I} (a_i \hat{x}_i + a_i^-) + b - \hat{\theta}. \end{aligned}$$

□

Using Proposition 4, we interpret the value $a_i \hat{x}_i + a_i^-$ as a measure of the importance of variable x_i for the cutting plane $\theta \geq \mathbf{a}^T \mathbf{x} + b$ at $\hat{\mathbf{x}}$. We use this intuition to construct a selection rule. We first pick a cutting plane $\theta \geq \mathbf{a}^T \mathbf{x} + b$ that approximates the epigraph of Q at $\hat{\mathbf{x}}$. Then indices $i \in [n]$ are added to the set I in decreasing order of the value $a_i \hat{x}_i + a_i^-$ until $|I| = K$. Note that $a_i \hat{x}_i + a_i^- \geq 0$ for any $a_i \in \mathbb{R}$ and $\hat{x}_i \in [0, 1]$. If the cutting plane approximation $\theta \geq \mathbf{a}^T \mathbf{x} + b$ is sparse (i.e., $|\{i \in [n] : a_i \neq 0\}|$ is small), it is possible that $|\{i \in [n] : a_i \hat{x}_i + a_i^- > 0\}| < K$. In such cases, we first add those indices with positive $a_i \hat{x}_i + a_i^-$ values into I , then pick another cutting plane and repeat the procedure until $|I| = K$. A potential advantage of this selection rule is that it does not require any evaluation of the cut violation function. And unlike the selection rule in Sect. 3.1, this selection rule can take advantage of the availability of dense cutting plane approximations. The potential limitation, of course, is the reliance on the single cutting-plane approximation.

The final detail we need to specify for this approach is how to choose the cutting-plane approximation(s). Assume a collection \mathcal{A} of cutting planes of the form $\theta \geq \mathbf{a}^T \mathbf{x} + b$ is available. A natural choice for \mathcal{A} is the set of cutting planes (e.g., Benders cuts) that have been added in the algorithm so far for approximating E^Q . A natural ordering for choosing which cutting plane in \mathcal{A} to use first is based on the tightness of the cutting plane at the point $\hat{\mathbf{x}}$. The inequality in \mathcal{A} with coefficients (\mathbf{a}, b) that yield the highest $\mathbf{a}^T \hat{\mathbf{x}} + b$ value is chosen first, etc.

4 Computational results

To provide insight into the computational potential of I -sparse cuts, we conduct numerical experiments on three MILP problems with block diagonal structures (2):

- The stochastic network interdiction (SNIP) problem [48] (40 instances): $n = 320$ for these instances.
- The latent-class logit assortment (LLA) problem [45] (36 instances): $n = 200$ for these instances.
- A stochastic version of the capacitated facility location (CAP) problem [14] (32 instances): n ranges between 25 and 50 for these instances.

We present the problem definition and details of the test instances for each problem in the Appendix. For the first two test problems, each block of their MILP formulations is sparse in variables \mathbf{x} , but in distinct ways. For the SNIP problem, we observe that when applying Benders decomposition to solve its LP relaxation the Benders cuts are mostly very sparse in \mathbf{x} . In the LLA problem each block of the MILP formulation only uses a small portion (between 12 and 20) of the \mathbf{x} variables, making the use of sparse cuts very natural for this problem. Neither of these two sparsity properties holds for the CAP problem.

In most of our test instances the constraints $\mathbf{x} \in X$ include binary restrictions on \mathbf{x} and either a lower or upper bound on the number of nonzero x_i variables. Therefore, we use $R(X) = \text{conv}(X)$ in those cases. In half of the LLA instances, X is a knapsack set with nonuniform coefficients and we use the direct polyhedral relaxation of X as $R(X)$. We use the direct LP relaxation as Q_k for each block of the MILP as described in Sect. 2.3.

We test the ability of I -sparse cuts to improve upon the standard LP relaxation within the Benders reformulation (3). The cut generating process is described in Algorithm 4. In the first step (line 1), we add standard Benders cuts iteratively until we have solved the initial LP relaxation. Specifically, this Benders approach works with a master LP relaxation in which the constraints $\theta_k \geq Q_k(\mathbf{x})$, $k \in [N]$ are approximated by Benders cuts of the form:

$$\theta_k \geq (\mu^j)^T (\mathbf{h}^k - T^k \mathbf{x}), \quad j = 1, \dots, t_k$$

where μ^j , $j = 1, \dots, t_k$, are extreme point solutions to the dual feasible region for subproblem k , $\Pi^k = \{\mu : \mu^T W^k \leq \mathbf{d}^k\}$. In the standard cutting-plane implementation [38], after solving a master LP relaxation and obtaining a solution $(\hat{\theta}, \hat{\mathbf{x}})$, Benders cuts

Algorithm 4: Generating I -sparse cuts

```

1 Initialize a master LP using Benders decomposition;
2 repeat
3   Solve the master LP to obtain solution  $(\hat{\theta}, \hat{\mathbf{x}})$ ;
4   for  $k \in [N]$  do
5     Choose a support  $I$ ;
6     Generate an  $I$ -sparse cut valid for the set  $E^{Q_k} = \{(\theta_k, \mathbf{x}) \in \mathbb{R} \times X : \theta_k \geq Q_k(\mathbf{x})\}$  by
       solving (12);
7     Add the  $I$ -sparse cut to the master LP if it is violated by  $(\hat{\theta}_k, \hat{\mathbf{x}})$ ;
8   end
9 until No violated cut can be generated or time limit is reached;

```

are identified by solving the subproblem (4) with $\mathbf{x} = \hat{\mathbf{x}}$ and adding the Benders cut defined by the dual optimal solution if it is violated by $(\hat{\theta}, \hat{\mathbf{x}})$. We use this standard cutting-plane method for the SNIP and LLA instances. We found the cutting-plane method took too long to converge for CAP instances, so we use the level method [40] for line 1 of Algorithm 4 on those instances.

In terms of the I -sparse cut generation, we consider the following variants of Algorithm 4:

- Greedy- K : Use the greedy rule described in Sect. 3.1 for generating the support I of size K ;
- Cutpl- K : Use the cutting plane approximation rule described in Sect. 3.2 for generating the support I of size K ;

We test Greedy- K and Cutpl- K with K fixed at 4, 7, and 10. We also test adaptive variants, Greedy-Ad and Cutpl-Ad of each selection method. These variants begin with $K = 4$. After solving the master LP, if $K < 10$ and the gap closed in the last five iterations for this K is less than 1% of the total gap closed thus far we increase K by 1 and re-start the generation of I -sparse cuts.

For Cutpl, we use the collection of all the Benders cuts added for block k in line 1 of Algorithm 4 as \mathcal{A} for Q_k . To improve the efficiency of the algorithm, when applying Greedy, we only select I from indices for which the corresponding variables have a nonzero coefficient in at least one of the Benders cuts for block k . This restriction is also implicitly implemented when using Cutpl since indices i with $a_i = 0$ for all $(\mathbf{a}, b) \in \mathcal{A}$ can never be selected by Cutpl. It significantly improves the efficiency of Greedy on SNIP instances (by skipping the generation of $\{i\}$ -sparse cuts for most $i \in [n]$).

All LPs and MILPs are solved using Gurobi 9.1.0. In the implementation of Algorithm 4, we take advantage of Gurobi's automatic warm start for solving similar LPs and store the values of $v_I^q(\chi)$ once evaluated for potential reuse in the future.

4.1 LP relaxation results

We first present results showing the impact of adding I -sparse inequalities to the LP relaxation of the problem without branching. An 1800-second time limit is set for generating I -sparse cuts in these experiments. To visually compare the performance

of I -sparse cuts across multiple test instances, we present results in the form of an integrality-gap-closed profile. Each curve in such a profile corresponds to a particular cut generation strategy, and its value at time t represents the average (over the set of instances for that problem class) integrality gap closed by time t , where the integrality gap closed at time t is calculated as $(z_R(t) - z_{LP}) / (z^* - z_{LP}) \times 100\%$, where $z_R(t)$ is the bound obtained by the algorithm at time t , z_{LP} is the basic LP relaxation bound, and z^* is the optimal value.

The results for the SNIP, LLA, and CAP test problems are given in Figs. 1, 2, and 3, respectively, where in each case we vary $K \in \{4, 7, 10\}$ or use an adaptively chosen K , and compare the Greedy and Cutpl selection rules. In each case we find that the

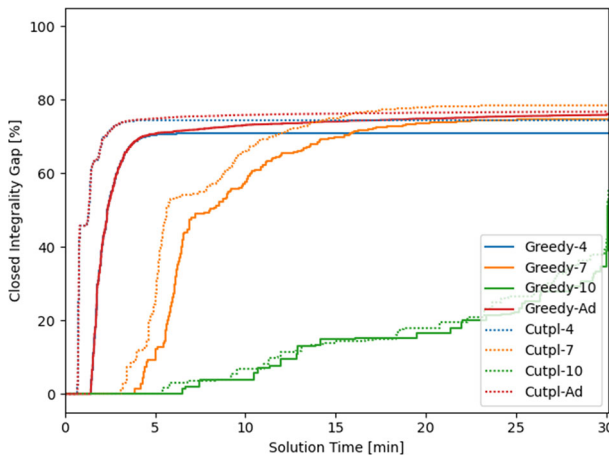


Fig. 1 Integrality-gap-closed profiles for SNIP instances obtained by different Greedy rules (solid) and Cutpl rules (dashed)

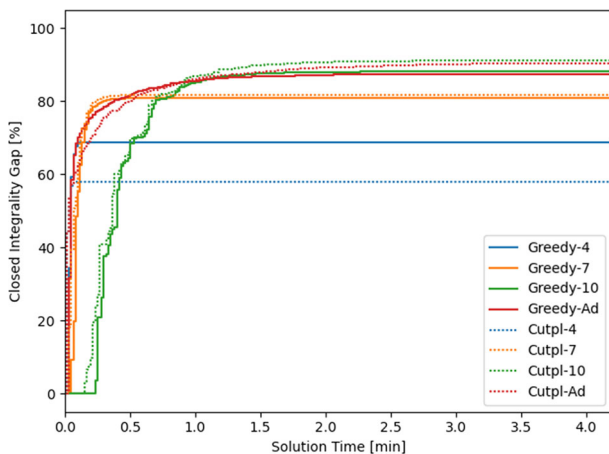


Fig. 2 Integrality-gap-closed profiles for LLA instances obtained by different Greedy rules (solid) and Cutpl rules (dashed)

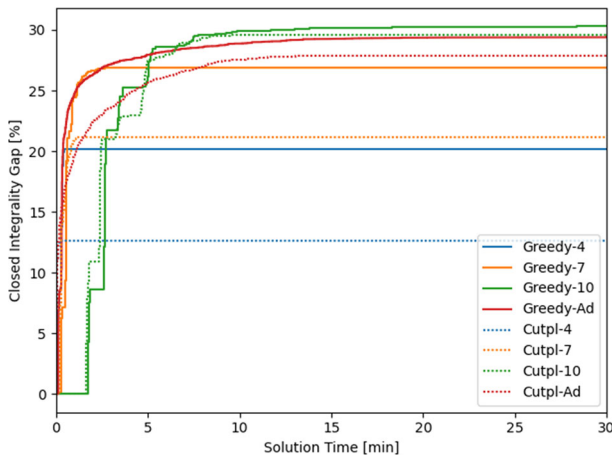


Fig. 3 Integrity-gap-closed profiles for CAP instances obtained by different Greedy rules (solid) and Cutpl rules (dashed)

two different selection rules have similar trends in gap closed over time. For fixed K , Cutpl rules perform better on the SNIP test instances, whereas Greedy rules have better performance on the LLA instances with $K = 4$ and the CAP instances. In terms of the effect of K , as expected smaller values of K yield quicker initial gap improvement, whereas larger values of K require more time to close the gap but eventually lead to more gap closed. For the SNIP instances we find that using $K = 4$ already closes most of the gap, and does so much more quickly than with $K = 7$ or $K = 10$. For the LLA instances we find that increasing K leads to more gap closed, although significant gap is already closed with $K = 4$, and the additional gap closed using $K = 10$ is marginal, while requiring significantly more time. For the CAP instances, we find that the I -sparse cuts close significantly less gap than the other test problems, although the gap closed is still significant. Large values of K yield significantly more gap closed on the CAP instances, but also requires considerably longer running time. The adaptive approaches Greedy-Ad and Cutpl-Ad appear to successfully achieve the best of the different choices of K , e.g., yielding quick improvement in bound early on while also eventually achieving bound improvement as good as achieved with the largest K .

For I -sparse cuts generated by different approaches, we summarize in Tables 1, 2, and 3 the closed integrity gap and number of cuts added. We observe that for the different fixed sparsity levels K , the number of I -sparse cuts added by the algorithm is not significantly larger for larger K . Thus, the improvement in the bound is attributable to stronger cuts rather than an increase in the number of cuts added. Naturally, the number of cuts added with the adaptive algorithm is larger than the number added for any fixed K , as that method potentially adds cuts for multiple values of K .

We next compare the I -sparse cuts with the multi-term 0-1 disjunctive cuts without the sparsity restriction, but generated from the same sets I , where the cuts are generated using the Perregaard and Balas (PB) [49] approach. Our interest in this comparison is to demonstrate the potential time reductions from using the I -sparse cuts and to estimate the extent to which the sparsity restriction degrades the quality of the relaxation. We

Table 1 LP relaxation results for SNIP instances. Results in each row are averages over five instances

Rule	snipno	b	Avg closed gap (%)				Avg # cuts added			
			4	7	10	Ad	4	7	10	Ad
Greedy	3	30	73.0	76.9	78.6	78.5	1030	932	895	1250
		50	71.6	75.5	63.8	76.8	1528	1365	706	1922
		70	73.5	78.1	46.6	79.5	1836	1574	396	2370
		90	67.4	71.4	34.9	72.0	2039	1715	326	2462
	4	30	72.3	77.0	79.0	79.0	1104	999	898	1831
		50	68.6	72.5	54.6	75.2	1520	1331	434	2019
		70	70.1	74.3	35.2	75.0	2063	1720	281	2599
		90	69.9	73.0	30.4	73.4	2454	2048	250	3517
Cutpl	3	30	76.7	78.6	78.6	77.7	1068	970	898	1185
		50	74.4	77.9	63.6	76.4	1361	1408	598	1776
		70	78.5	82.8	52.8	79.5	1471	1681	438	1722
		90	73.6	78.9	43.6	74.6	1631	1895	359	1826
	4	30	76.4	78.9	79.0	78.4	991	1046	886	1371
		50	72.6	77.0	55.8	73.5	1339	1286	399	1562
		70	73.5	78.4	39.6	74.6	1419	1446	290	1623
		90	69.6	75.7	31.6	70.3	1146	1200	255	1222

Table 2 LP relaxation results for LLA instances. Results in each row are averages over six instances

Rule	Cost	p	Avg closed gap (%)				Avg # cuts added			
			4	7	10	Ad	4	7	10	Ad
Greedy	Uniform	12	76.1	86.5	94.1	92.7	823	839	913	1960
		16	66.1	79.7	85.7	84.5	1129	1344	1430	3301
		20	63.9	74.3	82.0	82.3	1234	1400	1525	4069
	Nonunif	12	74.9	86.4	93.7	92.2	803	848	832	1744
		16	66.4	81.1	87.4	86.4	1097	1328	1440	3322
		20	64.0	75.3	83.9	84.2	1252	1432	1557	4132
Cutpl	Uniform	12	69.9	90.9	96.8	96.3	885	1127	1126	2834
		16	52.6	79.5	89.8	88.8	905	1493	1886	5420
		20	50.4	72.5	85.4	84.3	978	1604	1904	5778
	Nonunif	12	70.2	91.6	95.8	95.5	866	1078	991	2551
		16	52.8	80.8	90.1	89.4	944	1563	1881	5210
		20	50.1	73.3	86.5	85.7	946	1545	1971	5882

conduct this experiment only on the CAP test instances, since we have already seen that the I -sparse cuts are sufficient to close most of the gap in the SNIP and LLA instances, and thus there is little potential to close more gap when eliminating the sparsity restriction. We set a 24-hour time limit for the PB algorithm. For both the I -sparse and PB cuts, we use Greedy- K as the rule for selecting the set I to define the multi-term disjunction.

Table 3 LP relaxation results for CAP instances. Results in each row are averages over eight instances

Rule	Instance #	Avg closed gap (%)				Avg # cuts added			
		4	7	10	Ad	4	7	10	Ad
Greedy	101–104	34.6	44.4	51.8	47.0	790	848	732	2566
	111–114	7.3	13.6	15.7	15.0	518	770	560	2122
	121–124	14.6	20.1	21.9	21.2	647	659	576	1779
	131–134	24.3	29.6	31.7	31.1	757	732	659	1856
Cutpl	101–104	25.9	40.4	56.0	52.2	497	538	601	2074
	111–114	5.1	12.9	20.4	19.0	278	377	525	1921
	121–124	5.8	11.2	16.7	15.3	295	395	439	1337
	131–134	13.6	20.2	25.4	24.9	531	592	624	1708

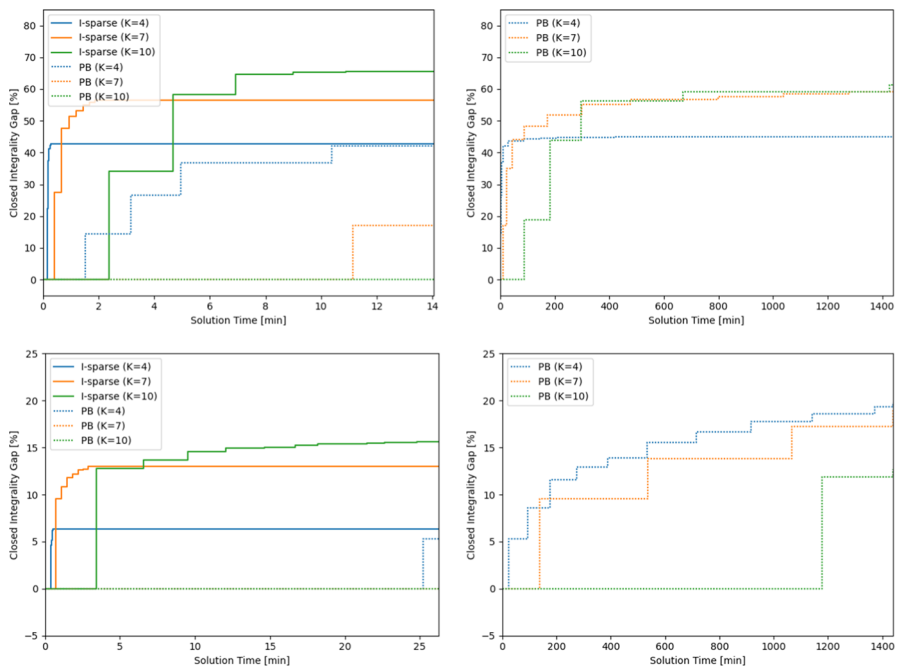
**Fig. 4** Integrality gap closed by *I*-sparse cuts and cuts generated by the PB algorithm on instances CAP101 (top) and CAP111 (bottom)

Figure 4 displays the integrality gap closed over time for two specific CAP instances, one for which *I*-sparse cuts were able to close a significant portion of the gap (CAP101), and one for which they were not (CAP111). The figures on the left display results for both the *I*-sparse cuts (solid lines) and PB cuts (dashed lines), with the time-scale (*x*-axis) determined by the time required to generate all *I*-sparse cuts for the largest value of *K*. From these figures we observe that for any value of *K*, within this time frame the *I*-sparse cuts close significantly more gap than the PB cuts. To

estimate the potential for PB cuts to eventually close more gap, we show the gap closed by the PB cuts over the full 24-hour time limit in the figures on the right. For CAP101 we find that the PB cuts do not close more gap than the I -sparse cuts, suggesting that the sparsity restriction is not significantly degrading the strength of the cuts in this case. On the other hand, for CAP111, we find that when given enough time the PB cuts can close significantly more gap, as seen particularly for the $K = 4$ results, although requiring far more time to do so. For both CAP instances, we observe that most of the generated PB cuts are as sparse as the I -sparse cuts in the first few iterations but become significantly denser (e.g., with non-zeros on more than half the variables) in later iterations.

4.2 Solving to optimality

We next present empirical results using I -sparse cuts within a branch-and-cut algorithm for exactly solving the test instances. The purpose of this study is to verify that the demonstrated relaxation improvement from these cuts translates to a reduction in the size of the search tree for these instances. We emphasize that our purpose is not to attempt to use these cuts to obtain state-of-the-art results, as such a test would require significant care in integrating multiple different types of cuts, etc.

We investigate using I -sparse cuts added at the root node to obtain an improved LP relaxation of the MILP (3), leading to a method we refer to as IBC (I -sparse branch-and-cut). We then solve the MILP instance, strengthened with the I -sparse cuts, via a Benders branch-and-cut algorithm. In this method, *Benders* cuts are added as lazy cuts at nodes in the branch-and-bound tree, as needed when integer feasible solutions are encountered. Specifically, when a solution $(\hat{\theta}, \hat{\mathbf{x}})$ with $\hat{\mathbf{x}}$ integer valued is obtained in the search process (either via a heuristic or as a solution of a node relaxation subproblem) we check whether it is feasible to (3), i.e., whether $\hat{\theta}_k \geq Q_k(\hat{\mathbf{x}})$ for all $k \in [N]$. If not we add Benders cuts as lazy constraints to cut off this infeasible solution and continue branch-and-cut. A more detailed description of Benders branch-and-cut can be found, e.g., in [14, 17]. We emphasize that we add I -sparse cuts only at the root node, so in terms of evaluation of the use of I -sparse cuts, this is a cut-and-branch approach. For generating I -sparse cuts, we use the Cutpl-Ad method from the previous section on the SNIP instances, and Greedy-Ad on the LLA and CAP instances. To achieve a balance between the benefit from the gap closed from the I -sparse cuts and the cut generation time, we terminate the cut generation process if the gap closed in the first two iterations of adding I -sparse cuts for a fixed K is smaller than 1% of the gap closed since the beginning of the I -sparse cut generation process. This choice is based on the empirical observation that the largest gap improvement almost always occurs in the first two iterations for each fixed K .

We compare against two other exact solution approaches, EXT — solving the MILP (2) directly in extensive form, and BBC — vanilla Benders branch-and-cut, which is identical to the IBC implementation except that the I -sparse cut generation step is skipped. A 3600-second time limit is set for solving each instance (including cut generation). Because the SNIP instances can be easily solved by BBC when solver cuts are used on top of the Benders cuts [14, 17], when solving the SNIP instances

Table 4 Root node results for SNIP instances. Results in each row are averages over five instances

snipno	b	I -sparse cut time (s)	LP int gap (%)		Root int gap (%)		Root time (s)	
			BBC	IBC	BBC	IBC	BBC	IBC
3	30	258	16.5	5.0	3.2	3.3	10	7
	50	553	21.8	6.5	4.7	4.7	12	9
	70	484	21.7	5.7	3.1	1.9	13	13
	90	473	24.5	8.2	4.8	4.0	14	13
4	30	379	19.4	5.3	3.7	2.9	11	10
	50	338	25.9	7.9	6.0	5.9	11	6
	70	432	27.9	8.0	4.5	4.0	15	11
	90	352	31.8	10.5	5.0	5.0	12	10

Table 5 Root node results for LLA instances. Results in each row are averages over six instances

Cost	p	I -sparse cut time (s)	LP int gap (%)		Root int gap (%)		Root time (s)	
			BBC	IBC	BBC	IBC	BBC	IBC
Uniform	12	59	5.3	0.7	3.9	0.5	5	3
	16	138	7.1	1.6	5.9	1.5	6	7
	20	168	6.9	1.9	6.1	1.8	7	8
Nonunif	12	54	5.2	0.7	4.9	0.6	6	4
	16	131	6.9	1.4	5.6	1.3	6	7
	20	186	6.8	1.6	5.9	1.5	8	9

Table 6 Root node results for CAP instances. Results in each row are averages of eight instances

Instance #	I -sparse cut time (s)	LP int gap (%)		Root int gap (%)		Root time (s)	
		BBC	IBC	BBC	IBC	BBC	IBC
101–104	405	18.3	11.7	17.4	10.8	2	3
111–114	811	8.1	6.9	7.7	6.7	7	6
121–124	586	17.8	14.8	17.0	14.3	8	7
131–134	548	21.3	16.6	20.2	15.8	5	6

using BBC or IBC, we turn off Gurobi presolve and cuts to show the impact of the I -sparse cuts. No changes to Gurobi's settings are made for the LLA or CAP instances, or for SNIP instances when solving with EXT.

We present in Tables 4, 5, and 6 the time spent on generating I -sparse cutting planes in IBC, the LP relaxation integrality gap,¹ the root node integrality gaps and the root node processing time of the Benders model associated with BBC and IBC. Unlike Sect. 4.2, Gurobi presolve and cuts are *not* turned off when generating these results for the

¹ Integrality gap of the lower bound LB is calculated as $(z^* - \text{LB})/|z^*| \times 100\%$ where z^* is the optimal objective value of the MILP.

Table 7 Exact solution results for SNIP instances. Time and gap results are averages over five instances

snipno	<i>b</i>	Avg soln time (s)			# solved instances			Avg opt gap (%)		
		EXT	BBC	IBC	EXT	BBC	IBC	EXT	BBC	IBC
3	30	≥3600	130	311	0/5	5/5	5/5	12.6	0.0	0.0
	50	≥3600	≥1569	638	0/5	4/5	5/5	19.0	0.5	0.0
	70	≥3600	≥3490	502	0/5	1/5	5/5	21.2	1.4	0.0
	90	≥3600	≥3600	644	0/5	0/5	5/5	20.8	5.0	0.0
4	30	≥3600	124	432	0/5	5/5	5/5	15.2	0.0	0.0
	50	≥3600	579	440	0/5	5/5	5/5	25.5	0.0	0.0
	70	≥3600	1651	539	0/5	5/5	5/5	31.3	0.0	0.0
	90	≥3600	1621	476	0/5	5/5	5/5	35.6	0.0	0.0

Table 8 Exact solution results for LLA instances. Time and gap results are averages over six instances

Cost	<i>p</i>	Avg soln time (s)			# solved instances			Avg opt gap (%)		
		EXT	BBC	IBC	EXT	BBC	IBC	EXT	BBC	IBC
Uniform	12	≥2666	≥3078	≥752	2/6	1/6	5/6	1.2	1.7	0.1
	16	≥3600	≥3600	≥3042	0/6	0/6	1/6	3.5	5.3	0.9
	20	≥3600	≥3600	≥3030	0/6	0/6	1/6	3.7	6.3	1.3
Nonunif	12	≥2390	≥2929	≥1231	3/6	2/6	4/6	1.1	1.5	0.1
	16	≥3600	≥3600	≥3035	0/6	0/6	1/6	3.2	4.9	0.8
	20	≥3600	≥3600	≥3031	0/6	0/6	1/6	3.5	6.5	1.1

Table 9 Exact solution results for CAP instances. Time and gap results are averages of eight instances

Instance #	Avg soln time (s)			# solved instances			Avg opt gap (%)		
	EXT	BBC	IBC	EXT	BBC	IBC	EXT	BBC	IBC
101–104	68	≥3600	≥3600	8/8	0/8	0/8	0.0	13.7	9.0
111–114	884	≥3600	≥3600	8/8	0/8	0/8	0.0	9.5	7.7
121–124	≥1313	≥3600	≥3600	7/8	0/8	0/8	<0.1	18.4	15.4
131–134	548	≥3600	≥3600	8/8	0/8	0/8	0.0	22.0	15.9

SNIP instances. We note that solver cuts successfully close the majority of integrality gap at the root node for SNIP instances, which is in line with the observations in [14, 17]. We also find that the LP relaxation bound of IBC is usually significantly stronger than that of BBC, which is consistent with the LP relaxation results in Sect. 4.1. In most cases, even though the root node processing can close more integrality gap in BBC than in IBC, IBC still achieves a smaller root node integrality gap, yielding a smaller search tree or a smaller optimality gap in the end. Additional root node results regarding the number of *I*-sparse cuts and solver cutting planes generated at the root node are presented in the Appendix.

Finally, we summarize the average solution time, number of solved instances and average ending optimality gap obtained by different methods on the SNIP instances, LLA instances, and CAP instances in Tables 7, 8, and 9, respectively. For the SNIP and LLA instances, we find that IBC either solves more instances and in less time, or yields smaller ending optimality gap than EXT and BBC, illustrating that the I -sparse cuts can indeed lead to improvements when solving these instances to optimality. EXT is the most effective method for solving the CAP instances, as observed also in [14]. However, in terms of Benders branch-and-cut based methods, the ending optimality gap of IBC is modestly smaller than BBC on these instances.

5 Extension: I -sparse tilting

We now describe an adaptation of the ideas used to create I -sparse inequalities to a more general setting by tilting a given valid inequality. Let $D \subseteq \{0, 1\}^n \times \mathbb{R}^p$ denote a mixed-binary set and let \mathbf{x} and \mathbf{y} denote the associated binary and continuous variables in the description of D , respectively. Let D^R be a relaxation of D with $D^R \cap (\{0, 1\}^n \times \mathbb{R}^p) = D$. We assume we are given a “base” inequality $\alpha^T \mathbf{x} + \beta^T \mathbf{y} \leq \gamma$ that is valid for D , and propose a method for “tilting” this inequality by modifying a subset of coefficients α_I of the binary variables, for $I \subseteq [n]$. Note that the I -sparse inequalities fit this more general setting by setting $p = 1$, $y_1 = \theta$, and using $-\theta \leq -\text{LB}$ as the base inequality, where LB is a lower bound on $Q(\mathbf{x})$ over $\mathbf{x} \in X$. Therefore, tilting leads to a broader class of valid inequalities, which may yield a stronger relaxation than the one generated by I -sparse inequalities when applied to the epigraph of $Q(\mathbf{x})$.

Consider a nonempty $I \subseteq [n]$ and the multi-term 0-1 disjunctive relaxation D_I^R of D defined as

$$D_I^R := \bigcup_{\chi \in \{0, 1\}^I} D_I^R(\chi),$$

where $D_I^R(\chi) := \{(\mathbf{x}, \mathbf{y}) \in D^R : \mathbf{x}_I = \chi\}$. We provide a sufficient condition for changes to the coefficients α_I in the inequality $\alpha^T \mathbf{x} + \beta^T \mathbf{y} \leq \gamma$ that assures the tilted inequality is valid for D .

Proposition 5 *Let $\alpha^T \mathbf{x} + \beta^T \mathbf{y} \leq \gamma$ be a valid inequality for D . Then an inequality $\mu^T \mathbf{x} + \beta^T \mathbf{y} \leq \eta$ with $\mu_{[n] \setminus I} = \alpha_{[n] \setminus I}$ is valid for D if*

$$\eta - \mu_I^T \chi \geq \min\{\bar{\nu}_I^R(\chi; \alpha, \beta), \gamma - \alpha_I^T \chi\}, \quad \forall \chi \in \{0, 1\}^I, \quad (21)$$

where $\bar{\nu}_I^R(\chi; \alpha, \beta) = \max\{\alpha_{[n] \setminus I}^T x_{[n] \setminus I} + \beta^T y : (\mathbf{x}, \mathbf{y}) \in D_I^R(\chi)\}$.

Proof Note that $\mu^T \mathbf{x} + \beta^T \mathbf{y} \leq \eta$ with $\mu_{[n] \setminus I} = \alpha_{[n] \setminus I}$ is valid for D if and only if it is valid for $\{(\mathbf{x}, \mathbf{y}) \in D : \mathbf{x}_I = \chi\}$ for all $\chi \in \{0, 1\}^I$, i.e.,

$$\max_{(\mathbf{x}, \mathbf{y}) \in D} \{\mu^T \mathbf{x} + \beta^T \mathbf{y} : \mathbf{x}_I = \chi\} \leq \eta, \quad \forall \chi \in \{0, 1\}^I. \quad (22)$$

Since $\alpha^T \mathbf{x} + \beta^T \mathbf{y} \leq \gamma$ is valid for D , $\beta^T \mathbf{y} \leq -\alpha^T \mathbf{x} + \gamma$ for all $(\mathbf{x}, \mathbf{y}) \in D$, and hence for all $\chi \in \{0, 1\}^I$ we have

$$\begin{aligned} \max_{(\mathbf{x}, \mathbf{y}) \in D} \{\mu^T \mathbf{x} + \beta^T \mathbf{y} : \mathbf{x}_I = \chi\} &\leq \max_{(\mathbf{x}, \mathbf{y}) \in D} \{\mu^T \mathbf{x} - \alpha^T \mathbf{x} + \gamma : \mathbf{x}_I = \chi\} \\ &= \max_{(\mathbf{x}, \mathbf{y}) \in D} \{(\mu_I - \alpha_I)^T \mathbf{x}_I + \gamma : \mathbf{x}_I = \chi\} \\ &= (\mu_I - \alpha_I)^T \chi + \gamma. \end{aligned}$$

On the other hand, for all $\chi \in \{0, 1\}^I$,

$$\begin{aligned} \max_{(\mathbf{x}, \mathbf{y}) \in D} \{\mu^T \mathbf{x} + \beta^T \mathbf{y} : \mathbf{x}_I = \chi\} &\leq \max_{(\mathbf{x}, \mathbf{y}) \in D_I^R(\chi)} \{\mu^T \mathbf{x} + \beta^T \mathbf{y}\} \\ &= \bar{v}_I^R(\chi; \alpha, \beta) + \mu_I^T \chi. \end{aligned}$$

Therefore,

$$\begin{aligned} \max_{(\mathbf{x}, \mathbf{y}) \in D} \{\mu^T \mathbf{x} + \beta^T \mathbf{y} : \mathbf{x}_I = \chi\} &\leq \min\{\bar{v}_I^R(\chi; \alpha, \beta) + \mu_I^T \chi, (\mu_I - \alpha_I)^T \chi + \gamma\} \\ &= \min\{\bar{v}_I^R(\chi; \alpha, \beta), \gamma - \alpha_I^T \chi\} + \mu_I^T \chi. \end{aligned} \quad (23)$$

Replacing $\max_{(\mathbf{x}, \mathbf{y}) \in D} \{\mu^T \mathbf{x} + \beta^T \mathbf{y} : \mathbf{x}_I = \chi\}$ in (22) by the upper bound in (23) yields the condition (21). \square

Note that $\alpha^T \mathbf{x} + \beta^T \mathbf{y} \leq \gamma$ does not need to be valid for D_I^R . But if $\alpha^T \mathbf{x} + \beta^T \mathbf{y} \leq \gamma$ is valid for D_I^R , then we can simply replace the minimum in (21) by $\bar{v}_I^R(\chi; \alpha, \beta)$ since $\bar{v}_I^R(\chi; \alpha, \beta) \leq \gamma - \alpha_I^T \chi$ for all $\chi \in \{0, 1\}^I$ in that case. Similar to (12), given a candidate solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ and fixed I , we can write down a cut generating linear program for separating from the inequalities satisfying (21):

$$\max \left\{ \sum_{i \in I} \mu_i \hat{x}_i - \eta : \eta - \mu_I^T \chi \geq \min\{\bar{v}_I^R(\chi; \alpha, \beta), \gamma - \alpha_I^T \chi\}, \chi \in \{0, 1\}^I \right\}.$$

We call any valid inequality $\mu^T \mathbf{x} + \beta^T \mathbf{y} \leq \eta$ satisfying (21) an I -sparse tilting of the inequality $\alpha^T \mathbf{x} + \beta^T \mathbf{y} \leq \gamma$. One may also iteratively tilt an inequality by applying an I_k -sparse tilting sequentially for some sequence $I_1, I_2, \dots \subseteq [n]$.

We next provide an example of obtaining valid inequalities using I -sparse tilting by showing that perspective cuts [28] for convex functions with indicator variables can be obtained by I -sparse tilting of subgradient inequalities for some I with $|I| = 1$.

Example 2 (Perspective cuts) The following mixed-binary structure is common in many applications with on/off decisions (e.g., the quadratic uncapacitated facility location problem [34]):

$$D = \{(x, \mathbf{z}, \theta) \in \{0, 1\} \times \mathbb{R}^{m+1} : \mathbf{0} \leq \mathbf{z} \leq \mathbf{u}x, \theta \geq f(\mathbf{z})\},$$

where $\mathbf{u} \in \mathbb{R}_+^m$ and $f : \mathbb{R}_+^m \rightarrow \mathbb{R}$ is a closed convex function with $f(\mathbf{0}) = 0$. The convex hull of F can be characterized by the perspective function of the function f [16]. One way of obtaining $\text{conv}(F)$ by the perspective function is to add the (potentially infinitely many) perspective cuts [28] of the form:

$$\theta \geq \mathbf{s}^T \mathbf{z} + (f(\bar{\mathbf{z}}) - \mathbf{s}^T \bar{\mathbf{z}})x, \quad (24)$$

where $\bar{\mathbf{z}} \in [\mathbf{0}, \mathbf{u}]$ and $\mathbf{s} \in \partial f(\bar{\mathbf{z}})$. We show that (24) can be obtained by I -sparse tilting of the following subgradient inequality of f :

$$\theta \geq f(\bar{\mathbf{z}}) + \mathbf{s}^T (\mathbf{z} - \bar{\mathbf{z}}) \Leftrightarrow 0x + \mathbf{s}^T \mathbf{z} - \theta \leq -f(\bar{\mathbf{z}}) + \mathbf{s}^T \bar{\mathbf{z}}. \quad (25)$$

By applying I -sparse tilting (Proposition 5) with $I = \{1\}$, the tilted inequality $\mu x + \mathbf{s}^T \mathbf{z} - \theta \leq \eta$ is valid if the following two inequalities hold:

1. $\eta \geq \min\{0, \mathbf{s}^T \bar{\mathbf{z}} - f(\bar{\mathbf{z}})\} = 0$ (because $0 = f(\mathbf{0}) \geq f(\bar{\mathbf{z}}) + \mathbf{s}^T (\mathbf{0} - \bar{\mathbf{z}})$);
2. $\eta - \mu \geq \min \left\{ \max\{\mathbf{s}^T \mathbf{z} - \theta : \theta \geq f(\mathbf{z}), \mathbf{0} \leq \mathbf{z} \leq \mathbf{u}\}, \mathbf{s}^T \bar{\mathbf{z}} - f(\bar{\mathbf{z}}) \right\} = \mathbf{s}^T \bar{\mathbf{z}} - f(\bar{\mathbf{z}})$ (because (25) attains equality at $(\mathbf{z}, \theta) = (\bar{\mathbf{z}}, f(\bar{\mathbf{z}}))$).

We then obtain the perspective cut (24) by choosing (μ, η) with both inequalities satisfied at equality, i.e., by setting $\mu = f(\bar{\mathbf{z}}) - \mathbf{s}^T \bar{\mathbf{z}}$ and $\eta = 0$.

6 Conclusion

We investigate methods for generating I -sparse cuts for the epigraph of a function of binary variables. Two selection rules are proposed to choose the support I . Numerical experiments demonstrate that I -sparse cuts are very effective on problems with sparse features.

We also extend our idea to strengthen valid inequalities by tilting coefficients on a sparse subset of variables. It would be interesting to explore this idea further. Another direction of future work is to integrate our techniques into SDDiP [57] to solve multistage stochastic integer programs.

Appendix

Test problems

SNIP problem

The SNIP problem [48] is a two-stage stochastic integer program with pure binary first-stage and continuous second-stage variables. In this problem, by installing sensors on some arcs of a directed network to in the first stage, the defender tries to find the attacker and minimize the probability that the attacker travels from the origin to the destination undetected. In the second stage, the origin and destination of the attacker are observed and the attacker chooses to travel on the maximum reliability path from

its origin to its destination. Let N and A denote the node set and the arc set of the network and let $D \subseteq A$ denote the set of interdictable arcs. The first-stage variables are denoted by \mathbf{x} , where $x_a = 1$ if and only if the defender installs a sensor on arc $a \in D$. Each scenario $s \in S$ is associated with a possible origin/destination combination of the attacker, with u^s representing the origin and v^s representing the destination of the attacker for each scenario $s \in S$. The second-stage variables are denoted by π , where π_i^s denotes the maximum probability of reaching destination v^s undetected from node i in scenario s . The budget for installing sensors is b , and the cost of installing a sensor on arc a is c_a for each arc $a \in D$. For each arc $a \in A$, the probability of traveling on arc a undetected is r_a if the arc is not interdicted, or q_a if the arc is interdicted. Parameter $\bar{\pi}_j^s$ denotes the maximum probability of reaching the destination undetected from node j when no sensors are installed. The extensive formulation of the problem is as follows:

$$\begin{aligned} \min_{\mathbf{x}, \pi^s} \quad & \sum_{s \in S} p_s \pi_{u^s}^s \\ \text{s.t.} \quad & \sum_{a \in A} c_a x_a \leq b, \\ & \pi_i^s - r_a \pi_j^s \geq 0, \quad a = (i, j) \in A \setminus D, s \in S, \\ & \pi_i^s - r_a \pi_j^s \geq -(r_a - q_a) \bar{\pi}_j^s x_a, \quad a = (i, j) \in D, s \in S, \\ & \pi_i^s - q_a \pi_j^s \geq 0, \quad a = (i, j) \in D, s \in S, \\ & \pi_{v^s}^s = 1, \quad s \in S, \\ & x_a \in \{0, 1\}, \quad a \in D. \end{aligned}$$

We use the SNIP instances from [48]. We consider 40 instances with $\text{snipno} \in \{3, 4\}$, and budget $b \in \{30, 50, 70, 90\}$. All instances have 320 first-stage binary variables, 2586 s-stage continuous variables per scenario and 456 scenarios.

LLA problem

The LLA problem is introduced in [45]. In this problem, a retailer chooses a set of items to display for customers to purchase. The model assumes all customers are from a set of customer segments. For each customer segment k , the customers arrive according to a Poisson process with rate λ_k and only purchases products in the consideration set C_k . For item i , the cost for displaying it is c_i , the relative attractiveness of it to customers in segment k is v_i^k , and the retailer earns a positive profit w_i for each purchase of it by the customers. The preference of not purchasing anything is denoted by v_0^k for customers in segment k . The retailer can choose items with a total cost up to $\rho \sum_{i=1}^n c_i$ to display. The problem can be formulated as an MINLP as follows:

$$\max_{\mathbf{x}} \left\{ \sum_{k=1}^N \sum_{i \in C_k} \frac{\lambda_k v_i^k w_i x_i}{\sum_{i \in C_k} v_i^k x_i + v_0^k} : \sum_{i=1}^n c_i x_i \leq \rho \sum_{i=1}^n c_i, \mathbf{x} \in \{0, 1\}^n \right\}.$$

In [45], the authors reformulate the MINLP as an MILP by introducing a variable y_k to represent the value of $1/(\sum_{i \in C_k} v_i^k x_i + v_0^k)$ and a variable z_{ik} to linearize the product $x_i y_k$. The MILP reformulation of the problem is as follows:

$$\begin{aligned}
 \max_{\mathbf{x}, \mathbf{y}, \mathbf{z}} \quad & \sum_{k=1}^N \sum_{i \in C_k} \lambda_k v_i^k w_i z_{ik} \\
 \text{s.t.} \quad & v_0^k y_k + \sum_{i \in C_k} v_i^k z_{ik} = 1, \quad k \in [N], \\
 & v_0^k y_k - v_0^k z_{ik} \leq 1 - x_i, \quad k \in [N], i \in C_k, \\
 & z_{ik} \leq y_k, \quad k \in [N], i \in C_k, \\
 & (v_0^k + v_i^k) z_{ik} \leq x_i, \quad k \in [N], i \in C_k, \\
 & \sum_{i=1}^n c_i x_i \leq \rho \sum_{i=1}^n c_i, \\
 & \mathbf{x} \in \{0, 1\}^n, \mathbf{y} \geq \mathbf{0}, \mathbf{z} \geq \mathbf{0}.
 \end{aligned}$$

For the generation of test instances, we follow the basic scheme of generating type-1 problems in [45], but increase the value of some of the parameters to make the instances harder. We set $n = 200$ and $N = 200$. For each segment k , its arrival rate λ_k and the preference of no purchase v_0^k are randomly generated according to the uniform distributions $\text{Uniform}([0, 1])$ and $\text{Uniform}([0, 4])$, respectively. The preference v_i^k for segment k of purchasing product i is randomly generated according to the discrete uniform distribution $\text{Uniform}(\{0, 1, \dots, 10\})$. For each odd $k \in K$, C_k is a independently randomly chosen subset of $[n]$ with size $p \in \{12, 16, 20\}$. For each even $k \in [N]$, C_k is a random subset of C_{k-1} of size $p/2$. The profit w_i of product i is independently randomly generated according to the uniform distribution $\text{Uniform}([100, U])$ with $U \in \{150, 350\}$. The capacity parameter ρ is chosen from $\{20\%, 50\%, 100\%\}$. We consider both cases with uniform and nonuniform costs for generating the cost parameters c_i :

(Uniform cost) $c_i = 1$ for $i \in [n]$;

(Nonuniform cost) $c_i = f_i \cdot |\{k \in [N] : i \in C_k\}|$ for $i \in [n]$, where each f_i is independently randomly generated according to the normal distribution with mean 1 and standard deviation 0.1.

We generate one instance with uniform costs and one instance with nonuniform costs for each combination of (p, U, ρ) . Therefore, we have 36 LLA instances in total.

CAP problem

The stochastic CAP problem [14] is a generalization of the deterministic CAP problem [43], which can be formulated as a stochastic two-stage integer program. In this problem, the decision maker chooses to open a set of facilities to meet uncertain customer demands. The first-stage variables are denoted by \mathbf{x} with $x_i = 1$ if and only if facility

i is chosen to be opened. The second-stage variables are denoted by \mathbf{y} , where y_{ij}^k is the amount of the j th customer's demand met by facility i in scenario k . For each facility i , the associated opening cost and its capacity are denoted by f_i and s_i , respectively. The cost associated with satisfying a unit of the j th customer demand using facility i (sending a unit of flow from facility i to customer j) is denoted by q_{ij} . The j th customer's demand under scenario k is denoted by λ_j^k . The extensive formulation of the problem is as follows:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}} \quad & \sum_{i=1}^n f_i x_i + N^{-1} \sum_{k=1}^N \sum_{i=1}^n \sum_{j=1}^m q_{ij} y_{ij}^k \\ \text{s.t.} \quad & \sum_{i=1}^n y_{ij}^k \geq \lambda_j^k, \quad j \in [m], k \in [N], \\ & \sum_{j=1}^m y_{ij}^k \leq s_i x_i, \quad i \in [n], k \in [N], \\ & \sum_{i=1}^n s_i x_i \geq \max_{k \in [N]} \sum_{j=1}^m \lambda_j^k, \\ & \mathbf{x} \in \{0, 1\}^n, \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

There are in total 32 CAP test instances all taken from [14] with $n \in \{25, 50\}$, $m = 50$ and $N = 250$.

Additional root node results

To provide further insights into the Benders model at the root node, in Tables 10, 11, and 12, we present the number of I -sparse cuts generated in IBC and the numbers of three major types of solver cutting planes (MIR, flow cover and relax-and-lift) generated in BBC and IBC, as reported by Gurobi. It is observed that a smaller

Table 10 Number of cutting planes generated for SNIP instances. Results in each row are averages over five instances

snipno	b	# I -sparse (in IBC)	# MIR		# Flow cover		# Relax-and-lift	
			BBC	IBC	BBC	IBC	BBC	IBC
3	30	1185	1007	198	0	0	179	49
	50	1776	646	292	284	0	221	57
	70	1722	1344	601	0	0	364	136
	90	1826	1244	617	198	137	279	93
4	30	1371	1045	286	0	0	244	46
	50	1562	1357	394	74	83	396	95
	70	1623	1294	448	0	0	481	124
	90	1222	1660	662	0	0	507	128

Table 11 Number of cutting planes generated for LLA instances. Results in each row are averages over six instances

Cost	p	# I -sparse (in IBC)	# MIR		# Flow cover		# Relax-and-lift	
			BBC	IBC	BBC	IBC	BBC	IBC
Uniform	12	2834	834	161	1224	166	4	0
	16	5420	908	434	1310	432	9	0
	20	5778	781	342	1085	372	8	0
Nonunif	12	2551	845	179	1125	121	6	1
	16	5210	923	324	1258	318	3	0
	20	5882	912	354	1129	366	9	0

Table 12 Number of cutting planes generated for CAP instances. Results in each row are averages of eight instances

Instance #	# I -sparse (in IBC)	# MIR		# Flow cover		# Relax-and-lift	
		BBC	IBC	BBC	IBC	BBC	IBC
101–104	2074	0	0	0	0	0	0
111–114	1921	1011	392	633	621	34	4
121–124	1337	762	263	718	531	10	2
131–134	1708	692	251	990	635	0	0

number of solver cuts are generated in IBC than in BBC. This is due to the fact that the LP relaxation of the Benders model in IBC is stronger than that of the Benders model in BBC. Consequently, even though the LP relaxation of IBC includes a great number I -sparse cuts, processing one node in the branch-and-bound tree is not significantly more challenging in IBC than in BBC, as fewer solver cutting planes are introduced there.

References

1. Ahmed, S.: Two-Stage Stochastic Integer Programming: A Brief Introduction Wiley Encyclopedia of Operations Research and Management Science. Wiley, USA (2010)
2. Andersen, K., Louveaux, Q., Weismantel, R., Wolsey, L.: Cutting planes from two rows of a simplex tableau. In: IPCO 2007: Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science, vol. 4513, pp. 1–15. Springer (2007)
3. Atamtürk, A., Nemhauser, G.L., Savelsbergh, M.W.: The mixed vertex packing problem. Math. Progr. **89**(1), 35–53 (2000)
4. Balas, E.: Disjunctive programming. In: Annals of Discrete Mathematics, vol. 5, pp. 3–51. Elsevier (1979)
5. Balas, E.: Disjunctive programming: properties of the convex hull of feasible points. Discret. Appl. Math. **89**(1–3), 3–44 (1998)
6. Balas, E., Ceria, S., Cornuéjols, G.: A lift-and-project cutting plane algorithm for mixed 0–1 programs. Math. Progr. **58**(1), 295–324 (1993)
7. Balas, E., Ceria, S., Cornuéjols, G.: Mixed 0–1 programming by lift-and-project in a branch-and-cut framework. Manage. Sci. **42**(9), 1229–1246 (1996)
8. Balas, E., Kazachkov, A.M.: V-polyhedral disjunctive cuts. arXiv preprint [arXiv:2207.13619](https://arxiv.org/abs/2207.13619) (2022)

9. Balas, E., Perregaard, M.: Lift-and-project for mixed 0–1 programming: recent progress. *Discret. Appl. Math.* **123**(1–3), 129–154 (2002)
10. Basu, A., Conforti, M., Cornuéjols, G., Zambelli, G.: Maximal lattice-free convex sets in linear subspaces. *Math. Oper. Res.* **35**(3), 704–720 (2010)
11. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* **4**, 238–252 (1962)
12. Bertsimas, D., Cory-Wright, R., Pauphilet, J.: A unified approach to mixed-integer optimization problems with logical constraints. *SIAM J. Optim.* **31**(3), 2340–2367 (2021)
13. Bertsimas, D., Tsitsiklis, J.N.: *Introduction to Linear Optimization*, vol. 6. Athena Scientific Belmont, MA (1997)
14. Bodur, M., Dash, S., Günlük, O., Luedtke, J.: Strengthened Benders cuts for stochastic integer programs with continuous recourse. *Inform. J. Comput.* **29**(1), 77–91 (2017)
15. Ceria, S., Pataki, G.: Solving integer and disjunctive programs by lift and project. In: R. Bixby, E. Boyd, R. Ríos-Mercado (eds.) *IPCO 1998: Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, vol. 1412, pp. 271–283. Springer (1998)
16. Ceria, S., Soares, J.: Convex programming for disjunctive convex optimization. *Math. Progr.* **86**(3), 595–614 (1999)
17. Chen, R., Luedtke, J.: On generating Lagrangian cuts for two-stage stochastic integer programs. *Inform. J. Comput.* **34**(4), 2332–2349 (2022)
18. Chen, R., Luedtke, J.: Sparse multi-term disjunctive cuts for the epigraph of a function of binary variables. In: K. Aardal, L. Sanitá (eds.) *IPCO 2022: Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, vol. 13265, pp. 98–111. Springer (2022)
19. Chvátal, V., Cook, W., Espinoza, D.: Local cuts for mixed-integer programming. *Math. Progr. Comput.* **5**(2), 171–200 (2013)
20. Cook, W., Kannan, R., Schrijver, A.: Chvátal closures for mixed integer programming problems. *Math. Progr.* **47**(1), 155–174 (1990)
21. Cornuéjols, G., Fisher, M.L., Nemhauser, G.L.: Exceptional paper-location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Manage. Sci.* **23**(8), 789–810 (1977)
22. Cornuéjols, G., Margot, F.: On the facets of mixed integer programs with two integer variables and two constraints. *Math. Progr.* **120**(2), 429–456 (2009)
23. Dey, S.S., Iroume, A., Molinaro, M.: Some lower bounds on sparse outer approximations of polytopes. *Oper. Res. Lett.* **43**(3), 323–328 (2015)
24. Dey, S.S., Lodi, A., Tramontani, A., Wolsey, L.A.: Experiments with two row tableau cuts. In: F. Eisenbrand, F. Shepherd (eds.) *IPCO 2010: Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, vol. 6080, pp. 424–437. Springer (2010)
25. Dey, S.S., Molinaro, M., Wang, Q.: Approximating polyhedra with sparse inequalities. *Math. Progr.* **154**(1), 329–352 (2015)
26. Dey, S.S., Molinaro, M., Wang, Q.: Analysis of sparse cutting planes for sparse MILPs with applications to stochastic MILPs. *Math. Oper. Res.* **43**(1), 304–332 (2018)
27. Espinoza, D., Fukasawa, R., Goycoolea, M.: Lifting, tilting and fractional programming revisited. *Oper. Res. Lett.* **38**(6), 559–563 (2010)
28. Frangioni, A., Gentile, C.: Perspective cuts for a class of convex 0–1 mixed integer programs. *Math. Progr.* **106**(2), 225–236 (2006)
29. Fukasawa, R., Poirrier, L., Yang, S.: Split cuts from sparse disjunctions. *Math. Progr. Comput.* **12**, 295–335 (2020)
30. Gade, D., Küçükyavuz, S., Sen, S.: Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Math. Progr.* **144**(1–2), 39–64 (2014)
31. Gray, F.: Pulse code communication. United States Patent Number 2632058 (1953)
32. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* **1**(2), 169–197 (1981)
33. Grötschel, M., Lovász, L., Schrijver, A.: *Geometric Algorithms and Combinatorial Optimization*. Springer Science & Business Media, Cham (2012)
34. Günlük, O., Lee, J., Weismantel, R.: MINLP Strengthening for Separable Convex Quadratic Transportation-Cost ufl. Tech. rep, IBM (2007)
35. Günlük, O., Pochet, Y.: Mixing mixed-integer inequalities. *Math. Progr.* **90**(3), 429–457 (2001)
36. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2021). <https://www.gurobi.com>

37. Kazachkov, A.M., Balas, E.: Monoidal strengthening of simple V-polyhedral disjunctive cuts. In: International Conference on Integer Programming and Combinatorial Optimization, pp. 275–290. Springer (2023)
38. Kelley, J.E., Jr.: The cutting-plane method for solving convex programs. *J. Soc. Ind. Appl. Math.* **8**(4), 703–712 (1960)
39. Laporte, G., Louveaux, F.V.: The integer L-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.* **13**(3), 133–142 (1993)
40. Lemaréchal, C., Nemirovskii, A., Nesterov, Y.: New variants of bundle methods. *Math. Progr.* **69**(1–3), 111–147 (1995)
41. Li, H.L.: A global approach for general 0–1 fractional programming. *Eur. J. Oper. Res.* **73**(3), 590–596 (1994)
42. Li, Y., Richard, J.P.P.: Cook, Kannan and Schrijver’s example revisited. *Discret. Optim.* **5**(4), 724–734 (2008)
43. Louveaux, F.V.: Discrete stochastic location models. *Ann. Oper. Res.* **6**(2), 21–34 (1986)
44. Lovász, L.: Submodular functions and convexity. In: *Mathematical Programming The State of the Art*, pp. 235–257. Springer, Cham (1983)
45. Méndez-Díaz, I., Miranda-Bront, J.J., Vulcano, G., Zabala, P.: A branch-and-cut algorithm for the latent-class logit assortment problem. *Discret. Appl. Math.* **164**, 246–263 (2014)
46. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions-I. *Math. Progr.* **14**(1), 265–294 (1978)
47. Ntaimo, L.: Fenchel decomposition for stochastic mixed-integer programming. *J. Global Optim.* **55**(1), 141–163 (2013)
48. Pan, F., Morton, D.P.: Minimizing a stochastic maximum-reliability path. *Networks* **52**(3), 111–119 (2008)
49. Perregaard, M., Balas, E.: Generating cuts from multiple-term disjunctions. In: K. Aardal, B. Gerards (eds.) *IPCO2001: Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, vol. 2081, pp. 348–360. Springer (2001)
50. Rahmiani, R., Ahmed, S., Crainic, T.G., Gendreau, M., Rei, W.: The Benders dual decomposition method. *Oper. Res.* **68**, 878–895 (2020)
51. Rockafellar, R.T.: *Convex Analysis*. 28. Princeton university press (1970)
52. Sen, S., Hige, J.L.: The C3 theorem and a D2 algorithm for large scale stochastic mixed-integer programming: set convexification. *Math. Progr.* **104**(1), 1–20 (2005)
53. Sen, S., Sherali, H.D.: Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Math. Progr.* **106**(2), 203–223 (2006)
54. Van Slyke, R.M., Wets, R.: L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM J. Appl. Math.* **17**(4), 638–663 (1969)
55. Wiegele, A.: *Biq Mac Library—a collection of max-cut and quadratic 0-1 programming instances of medium size*. Tech. rep. (2007)
56. Zhang, M., Küçükyavuz, S.: Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs. *SIAM J. Optim.* **24**(4), 1933–1951 (2014)
57. Zou, J., Ahmed, S., Sun, X.A.: Stochastic dual dynamic integer programming. *Math. Progr.* **175**(1–2), 461–502 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.