Noname manuscript No.
(will be inserted by the editor)

A Finitely Convergent Cutting Plane, and a Bender's Decomposition Algorithm for Mixed-Integer Convex and Two-Stage Convex programs using Cutting Planes

Fengqiao Luo · Sanjay Mehrotra

Received: date / Accepted: date

Abstract We consider a general mixed-integer convex program. We first develop an algorithm for solving this problem, and show its finite convergence. We then develop a finitely convergent decomposition algorithm that separates binary variables from integer and continuous variables. The integer and continuous variables are treated as second stage variables. An oracle for generating a parametric cut under a subgradient decomposition assumption is developed. The decomposition algorithm is applied to show that two-stage (distributionally robust) convex programs with binary variables in the first stage can be solved to optimality within a cutting plane framework. For simplicity, the paper assumes that certain convex programs generated in the course of the algorithm are solved to optimality.

1 Introduction

We consider a family of mixed-integer convex programs in the form:

$$\begin{aligned} \min\ & g_0(x,y) \\ \text{s.t.}\ & g_i(x,y) \le 0 \quad \forall i \in \mathcal{I}, \\ & x \in \mathcal{X} \cap \{0,1\}^{l_1},\ y \in \mathcal{Y} \cap (\mathbb{Z}^{l_2} \times \mathbb{R}^{l_3}), \end{aligned} \tag{JMICP}$$

where x are binary variables, and y are mixed-integer variables. The functions $g_0(x,y)$ and $g_i(x,y)$ for $i \in \mathcal{I}$ are convex, and $\mathcal{X}$ and $\mathcal{Y}$ are bounded convex sets

Fengqiao Luo
Department of Industrial Engineering and Management Science, Northwestern University
Evanston, Illinois
fengqiaoluo2014@u.northwestern.edu

Sanjay Mehrotra, Corresponding author
Department of Industrial Engineering and Management Science, Northwestern University
Evanston, Illinois
mehrotra@northwestern.edu

for x and y. An important special case of (JMICP) is the following two-stage stochastic mixed-integer convex program:

$$\begin{aligned} \min_{x}\ & c(x) + \mathbb{E}_{\xi\sim P\in\mathcal{P}}[\mathcal{Q}(x,\xi)] \\ \text{s.t.}\ & Ax \le b,\ x\in\mathcal{C}, \\ & x\in\{0,1\}^{l_1}, \end{aligned} \tag{1}$$

where x is the vector of first-stage variables that are pure binary, and ξ is a vector of random parameters that follow the joint distribution P. The recourse function $\mathcal{Q}(x,\xi^\omega)$ for scenario $\omega\in\Omega$ is given as:

$$\begin{aligned} \mathcal{Q}(x,\xi^\omega) = \min_{y^\omega}\ & q^{\omega\top}y^\omega \\ \text{s.t.}\ & g_j(s_j^{\omega\top}y^\omega + t_j^{\omega\top}x + r_j^\omega) \le 0 \qquad \forall j\in[J], \\ & y^\omega\in\mathbb{Z}^{l_2}\times\mathbb{R}^{l_3}, \end{aligned} \tag{2}$$

where every g_j is a convex function, and (q^ω, s_j^ω, t_j^ω, r_j^ω) are scenario-specific coefficients.

In addition to developing an algorithm for solving mixed integer convex programs, a major result of this paper is to develop a finitely convergent cutting-plane decomposition algorithm for solving (JMICP) under mild regularity conditions. Parametric cuts are generated within the decomposition algorithm. An oracle for generating the parametric cuts is also developed. The decomposition algorithm makes the following assumptions.

Assumption 1 *For any fixed $x\in\mathcal{X}\cap\{0,1\}^{l_1}$,* (JMICP) *is feasible in y.*

Assumption 2 *For any $[x_0,y_0]\in\mathbb{R}^{l_1+l_2+l_3}$, the function g_i $\forall i\in\mathcal{I}_0\cup\{0\}$ satisfies $\partial g_i(x_0,y_0) = \partial_x g_i(x_0,y_0)\times\partial_y g_i(x_0,y_0)$, where $\partial g_i(x_0,y_0)$ is the set of sub-gradients of g_i at the point $[x_0,y_0]$, $\partial_x g_i(x_0,y_0)$ is the set of sub-gradients of the function $g_i(x,y_0)$ at $x=x_0$, and $\partial_y g_i(x_0,y_0)$ is defined similarly.*

Assumption 3 *We have an oracle that can solve a general convex optimization problem to optimality without error.*

Assumptions 1, 3 simplify our presentation and analysis. It is possible to add additional artificial variables to ensure Assumption 1. Assumption 2 is used when ensuring that the cutting planes generated to solve (JMICP) in the space of y for any fixed x are valid and extendable to generate a Bender's cut in the space of x. More detailed explanation regarding this assumption is provided at the end of Section 4.1. Note that Assumption 2 is satisfied by two important families of functions: (1) The differentiable convex functions in the $[x,y]$-space; (2) The separable convex functions in the form of $g(x,y)=\psi(x)+\phi(y)$, where ψ and ϕ are convex. This is the case in two-stage stochastic convex programs. Assumption 3 is needed to prove the finite convergence of our algorithm which frequently calls the oracle to solve a general convex optimization problem to optimality in a finite number of iterations. It is possible to develop a variant of the algorithm presented here that allows ϵ-error in solving convex programs. However, it is not presented here due to space limitations.

In the context of two-stage stochastic convex programs, we assume that Ω is a finite set. In the case where the set $\mathcal{P}$ is singleton, i.e., there is no distributional ambiguity, (1)–(2) also admits an extended reformulation in terms of variables x and y^ω for $\omega \in \Omega$:

$$\begin{aligned} \min\ & c(x) + \sum_{\omega\in\Omega} p^\omega q^{\omega\top} y^\omega \\ \text{s.t.}\ & g_j(s_j^{\omega\top} y^\omega + t_j^{\omega\top} x + r_j^\omega) \le 0 \qquad \forall j \in [J], \\ & Ax \le b,\ x \in \mathcal{C}, \\ & x \in \{0,1\}^n,\ y^\omega \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2}, \end{aligned} \tag{3}$$

where p^ω is the probability of the scenario ω. It is straightforward to verify that (3) is in the form of (JMICP). Thus (3) can be solved directly using an algorithm for solving mixed integer convex programs. However, even in this case, a Bender's decomposition approach is typically preferable when the number of scenarios is large.

1.1 Literature Review

The development of methods and algorithms for solving mixed-integer convex programs (MICPs) benefits from more mature solvers for mixed-integer linear programs (MILPs). We provide a brief review on work for solving MILPs with cutting planes, the outer approximation approach for solving MICPs based on cutting-plane methods, and generalization of these methods for solving two-stage stochastic mixed integer programs.

Branch-and-cut algorithms [1, 2] are well developed methods for solving a general MILP. The cutting planes are generated at the root relaxation node or some other node of the branch-and-bound tree to strengthen the linear relaxation, or cut the current solution having undesirable fractional components. Many families of valid inequalities have been developed since 1950's to serve as cutting planes in the branch-and-cut algorithm. In particular, we have general purpose cuts [3] such as Gomory cuts [4], disjunctive cuts [8], mixed-integer rounding cuts [**?**], and polyhedral structure based cuts such as flow cover inequalities [5] and flow-path inequalities [6]. Generation of these cuts plays an important role in a branch-and-cut algorithm. A natural question is whether a MILP can be solved to optimality by adding cutting planes only. For a general mixed 0-1 linear program, an affirmative answer is given in [7], in which a systematic way of adding disjunctive cuts to the linear relaxation problem is given. The disjunctive cuts are generated using the lift-and-project method investigated in [8, 9], which amounts to solving a linear program to obtain the coefficients of a disjunctive cut. Owen and Mehrotra [10] developed a cutting-plane algorithm for solving a general mixed-integer linear program. To guarantee algorithm's convergence to an optimal solution, cuts are added at all γ-optimal (a notion defined in [10]) vertices of the LP-relaxation master problem at every iteration. As an alternative to explring the γ-optimal vertices, Chen et. al. [12] proposed a convergent cutting-plane algorithm for solving a general mixed-integer linear program to optimality with a linear-program solver as the only oracle. The key idea is to build a tree to keep track of convex hulls formed by the union of disjunctive polytopes. Jörg [13] provided an alternative convergent cutting-plane algorithm that can solve a general MILP to optimality by introducing the notion of k-disjunctive cuts, which is a generalization of the disjunctive cuts given in [8, 9].

Outer approximation (OA) methods for solving a mixed integer convex program (MICP) also have a long history. The basic idea can be traced back to [14]. In an outer approximation method, the convex set is outer approximated with a polyhedron which relaxes the MICP with a MILP in every master iteration. The polyhedron approximation is progressively refined by adding more valid inequalities that are tangent to the convex set. The theoretical question on whether a general MICP can be solved to optimality with cutting planes is not well addressed, particularly when the convex functions are not differentiable. Towards addressing this question, Lubin et. al. [20] use a polishing step after solving the master MILP, which solves a continuous convex optimization problem by fixing the integer variables to be the values of integer part in the current master solution. This polishing step is similar to the solution polishing done within the algorithm described in [10]. The solution from the polishing-step problem is compared with the master problem solution to check whether an optimal solution is identified. However, the proof given in [20] is incomplete, and is not applicable for the case where the convex functions are not differentiable. The convex functions considered in [20] are assumed to be differentiable, in which case, the outer approximation cutting plane is unique at a specific boundary point and it can be obtained by taking gradient of a constraint function. For a MICP involving general convex functions, this approach requires generalization. In particular, if the convex set is a convex cone the number of tangent planes passing the origin can be infinite, in which case, how to add the cutting planes is unclear.

Cutting plane methods have also been used for solving two-stage stochastic mixed-integer linear programs (TSS-MILP) with mixed-integer second-stage variables. One approach is to generate parametric Gomory cuts that sequentially convexify the feasible set [21, 22]. Gade et al. [21] have shown the finite-convergence of this algorithm for solving TSS-MILPs with pure-binary first-stage variables and pure-integer second-stage variables based on generating Gomory cuts that are parameterized by the first-stage solution. This approach is generalized by Zhang and Küçükyavuz [22] for solving TSS-MILP with pure-integer variables in both stages. Recent work has also provided insights into developing tighter formulations by identifying globally valid parametric inequalities (see [23], and references therein). For two-stage stochastic mixed-integer conic optimization, Bansal and Zhang [24] have developed nonlinear sparse cuts for tightening the second-stage formulation of a class of two-stage stochastic p-order conic mixed-integer programs by extending the results of [25] on convexifying a simple polyhedral conic mixed-integer set.

Sen and Sherali [26] developed a decomposition framework for solving a deterministic pure-binary linear program with two sets of binary decision variables coupled by some linking constraints. The algorithm iteratively solves a master problem and a sub-problem, each involving different sets of variables. The Bender's cut added to the master problem is a disjunctive valid inequality generated via taking union of all leaf nodes in the branch-and-bound tree constructed for solving the sub-problem. This framework is applied to solve two-stage stochastic mixed-integer programs with pure binary first-stage variables, as shown in [26]. In [29] we showed that the decomposition framework with branch-and-cut approach can be further developed for the two-stage stochastic mixed-integer conic programming using the branch-and-union approach as well as an approach where the scenarios subproblems are solved using cutting-planes. The cutting plane framework presented in this paper can be further generalized to incorporate branching together with the addition of cuts using the framework from [29]. We leave it to a future exposition due to space constraints.

1.2 Contributions of this paper

This paper makes the following contributions:

- A cutting-plane algorithm is established for solving a general mixed-integer convex program (Section 2). It is proved that the algorithm can identify an optimal solution of the MICP in finitely many iterations with finitely many cutting planes (Section 3). To the best of our knowledge, this is the first formal proof of finitely convergence of the cutting-plane algorithm for solving a general MICP.
- A decomposition algorithm is developed for solving (JMICP). A novel aspect of this algorithm is an oracle that develops a parametric cut that is added to the master problem in the decomposition algorithm (Section 4).
- The cutting-plane algorithm is applied to solve a family of distributionally-robust two-stage stochastic mixed-integer convex programs where the first stage decision variables are pure binary. (Section 5).

Section 6 illustrates the decomposition algorithm using a numerical example.

2 An Algorithm for Mixed-Integer Convex programs using Cutting Planes

We first consider a mixed-integer convex program in the form:

$$\begin{aligned} &\min\ c^\top x \\ &\text{s.t.}\ \ Ax \le b, \quad x \in \mathcal{C},\ x \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2}, \end{aligned} \tag{MICP}$$

where $\mathcal{C}$ is a general convex set, and $Ax \leq b$ are linear constraints. Note that (MICP) is general enough to incorporate the case that the objective is a nonlinear convex function, as the nonlinear term can be reformulated into a constraint, leaving only linear terms in the objective afterwards. As a special case, the set $\mathcal{C}$ can be described by inequality constraints $c_i(x) \leq 0$ for $i \in [m]$, where $[m]$ represents the set of integers $\{1, \dots, m\}$, and $c_i(\cdot) : \mathbb{R}^{l_1+l_2} \to \mathbb{R}$ are proper convex functions. In another important special case, the set $\mathcal{C} := \cap_{t=1}^{T} \mathcal{K}^t$, where $\mathcal{K}^t$ are proper cones. The cutting plane algorithm developed in this section for solving (MICP) calls an oracle given in the following proposition.

Proposition 2.1 *For any general convex sets $S_k \subset \mathbb{R}^n$ with $k \in [K]$, any point $x \in bd(\cap_{k=1}^{K} S_k)$, and any vector c satisfying*

$$c \in \sum_{k=1}^{K} \mathcal{N}_{S_k}(x), \tag{4}$$

where $\mathcal{N}_{S_k}(x)$ is the normal cone of S_k at x, and the addition follows the Minkowski rule. There exists an oracle that can generate K points v_k for $k \in [K]$ (some of them can be zero) such that

$$c = \sum_{k=1}^{K} v_k \quad \textit{and} \quad v_k \in \mathcal{N}_{S_k}(x) \; \forall k \in [K]. \tag{5}$$

Proof Solving the following convex program can generate v_k $\forall k \in [K]$ with the desired property:

$$\min \sum_{k=1}^{K} \|v_k\|_2 \quad \text{s.t. } c = \sum_{k=1}^{K} v_k, \; v_k \in \mathcal{N}_{S_k}(x) \; \forall k \in [K]. \tag{6}$$

Remark 2.1 It is important to note that the cone $\mathcal{N}_{S_k}(x)$ may not be finitely generated. Although the normal cone $\mathcal{N}_{S_k}(x)$ is written in an abstract form in the proposition, we assume that it is available in an analytical form so that the convex optimization problem (6) can be solved numerically. To give an example, assume that $K = 2$, $S_1 = \{[x, t] : \|A_1 x\|_2 \leq b_1 t\}$ and $S_2 = \{[x, t] : \|A_2 x\|_2 \leq b_2 t\}$, where $[x, t] \in \mathbb{R}^n \times \mathbb{R}$ and $c_1, c_2 > 0$. Consider the normal cone of S_1 and S_2 at $y = [\mathbf{0}, 0]$, which can be $\mathcal{N}_{S_1} = \{[x, t] : \|(A_1^{-1})^\top x\| \leq -t/b_1\}$ and $\mathcal{N}_{S_2} = \{[x, t] : \|(A_2^{-1})^\top x\| \leq -t/b_2\}$. Then the optimization problem (6) becomes the following second-order cone program:

$$\begin{aligned} \min \; & \|[x_1, t_1]\|_2 + \|[x_2, t_2]\|_2 \\ \text{s.t. } & [x_1 + x_2, t_1 + t_2] = c, \\ & \|(A_1^{-1})^\top x_1\| \leq -t_1/b_1, \\ & \|(A_2^{-1})^\top x_2\| \leq -t_2/b_2. \end{aligned}$$

In the above example, for simplicity, we have kept both second order cones be pointed at the origin. However, in the proposition allows for these cones be pointed at different points; and thus not requiring a translation of cones not pointed at the origin to the origin by adding new equality constraints.

Definition 2.1 For a general convex set S, an inequality $\alpha^\top x + \beta \leq 0$ is called a supporting valid inequality for S if it satisfies

1. $\alpha^\top x + \beta \leq 0$ for all $x \in S$,
2. and $\alpha^\top \hat{x} + \beta = 0$ for an $\hat{x} \in S$, i.e., it is a supporting hyperplane of S at $\hat{x}$.

We now provide an overview of our algorithm. A pseudo-code is given in Algorithm 1. We will show that the algorithm terminates with an optimal solution after adding finitely many cuts. At the main iteration n we consider the following mixed-integer linear program as a master problem of the current iteration:

$$\begin{aligned} \min_{x \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2}} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b, \\ & (x^{(k)} - z^{(k)})^\top x \leq (x^{(k)} - z^{(k)})^\top z^{(k)} \qquad \forall k \in [n-1], \\ & C^{(k)} x \leq d^{(k)} \quad \forall k \in \mathcal{I}^{(n-1)}, \end{aligned} \tag{MS-n}$$

where $(x^{(k)} - z^{(k)})^\top x \leq (x^{(k)} - z^{(k)})^\top z^{(k)}$ is the cutting plane generated at iteration k to separate the master problem solution $x^{(k)}$ from $\mathcal{C}$ at iteration k, and $C^{(k)} x \leq d^{(k)}$ are outer-approximation cuts generated to approximate $\mathcal{C}$ at iteration $k \in [n-1]$. Since the outer-approximation cuts are not necessarily available at every iteration, the algorithm uses the index set $\mathcal{I}^{(n-1)}$ to record the iteration indices in $[n-1]$ for which the outer-approximation cuts are generated. Generation of these cuts will become more clear as we explain the algorithm.

The algorithm solves (MS-n) by using the oracle from [12] for solving a general mixed-integer linear programm using cutting planes without branching. If the mixed-integer linear program is infeasible, we stop and conclude that (MICP) is infeasible. Otherwise, let $x^{(n)}$ be a solution of the master problem (MS-n). If $x^{(n)} \in \mathcal{C}$, then $x^{(n)}$ is optimal to (MICP) and the algorithm stops. Otherwise, consider the projection problem:

$$\min_{z \in \mathcal{C}} \left\| z - x^{(n)} \right\|_2^2 \tag{7}$$

and let $z^{(n)}$ be its optimal solution. Then the algorithm generates the following separation cut:

$$(x^{(n)} - z^{(n)})^\top x \leq (x^{(n)} - z^{(n)})^\top z^{(n)}. \tag{8}$$

The algorithm first adds (8) to (MS-n). Then it solves the following continuous convex program:

$$\begin{aligned} \min_{x \in \mathcal{C}} \ & c^\top x \\ \text{s.t.} \ & Ax \leq b, \\ & x_j = x_j^{(n)} \quad j \in [l_1]. \end{aligned} \tag{9}$$

The problem in (9) can be viewed as a solution polishing step. It also plays a key role in our finite convergence analysis. There are three cases: (a) The problem (9) is infeasible; (b) Find an optimal solution $\bar{x}^{(n)}$ of (9), and it is in the interior of $\mathcal{C}$, i.e., $\bar{x}^{(n)} \in \text{int}(\mathcal{C})$; (c) The optimal solution $\bar{x}^{(n)}$ is on the boundary of $\mathcal{C}$, i.e., $\bar{x}^{(n)} \in \text{bd}(\mathcal{C})$. The algorithm will proceed accordingly. For the cases (a) and (b), the algorithm sets $\mathcal{I}^{(n)} \leftarrow \mathcal{I}^{(n-1)}$, updates $n \leftarrow n+1$, and proceeds to the next iteration.

In the case (c), the optimality condition ensures that the objective vector c is in the normal cone $\mathcal{N}_{S^{(n)}}(\bar{x}^{(n)})$ for $S^{(n)} = \{x : Ax \leq b,\ x_j = x_j^{(n)}\ \forall j \in [l_1],\ x \in \mathcal{C}\}$. The algorithm runs the oracle given in Proposition 2.1 to obtain $r' \in \mathcal{N}_{S'^{(n)}}(\bar{x}^{(n)})$ and $r \in \mathcal{N}_{\mathcal{C}}(\bar{x}^{(n)})$ satisfying $c = r' + r$, where $S'^{(n)} = \{x : Ax \leq b,\ x_j = x_j^{(n)}\ \forall j \in [l_1]\}$. In this case, the algorithm constructs the following supporting linear inequality

$$r^\top (x - \bar{x}^{(n)}) \leq 0. \tag{10}$$

For simplicity, we rewrite the above inequality as the following system

$$C^{(n)} x \leq d^{(n)}. \tag{11}$$

By Lemma 3.3, the oracle ensures the following relation:

$$\begin{aligned} &\{\min c^\top x \mid Ax \leq b,\ x_j = x_j^{(n)}\ \forall j \in [l_1],\ x \in \mathcal{C}\} \\ &= \{\min c^\top x \mid Ax \leq b,\ x_j = x_j^{(n)}\ \forall j \in [l_1],\ C^{(n)} x \leq d^{(n)}\}. \end{aligned} \tag{12}$$

The additional cuts (11) are added to (MS-n). Update $\mathcal{I}^{(n)} \leftarrow \mathcal{I}^{(n-1)} \cup \{n\}$, $n \leftarrow n+1$, and proceed to the next iteration. Some key notations used in Algorithm 1 are given below:

- $x^{(k)}$: an optimal solution of the master problem at iteration k;
- $z^{(k)}$: the projection of $x^{(k)}$ on $\mathcal{C}$, computed at iteration k;
- $(x^{(k)} - z^{(k)})^\top x \leq (x^{(k)} - z^{(k)})^\top z^{(k)}$: the separation cut generated at iteration k;
- $\bar{x}^{(k)}$: an optimal solution of the continuous convex program $\left\{\min\ c^\top x \,\middle|\, Ax \leq b,\ x_j = x_j^{(k)}\ \forall j \in [l_1],\ x \in \mathcal{C}\right\}$ obtained at iteration k;
- $C^{(k)} x \leq d^{(k)}$: the supporting valid inequalities, generated at iteration k in the case that $\bar{x}^{(k)} \in \text{bd}(\mathcal{C})$;
- $\mathcal{I}^{(n)}$: this is the subset $\left\{k \in [n] \,\middle|\, \bar{x}^{(k)} \in \text{bd}(\mathcal{C}),\ c \in \mathcal{N}_{S^{(k)}}(\bar{x}^{(k)})\right\}$ of iteration indices for which the supporting valid inequalities are generated.

3 Convergence Analysis for the Cutting Plane Algorithm

We first provide a few technical results that are used for proving the main result (Theorem 3.1) in this section.

Proposition 3.1 *If $\{x^{(n)}\}_{n=1}^{\infty}$ is a convergent sequence in $\mathbb{Z}^{l_1} \times \mathbb{R}^{l_2}$, then the limit point of this sequence is also in $\mathbb{Z}^{l_1} \times \mathbb{R}^{l_2}$. There exists an index N and a vector $v \in \mathbb{Z}^{l_1}$ such that $x_i^{(n)} = v_i$ for all $i \in [l_1]$ and all $n \geq N$.*

Algorithm 1 An algorithm for solving a mixed integer convex program.

Set $n \leftarrow 1$, $flag \leftarrow 0$ and $\mathcal{I}^{(0)} \leftarrow \emptyset$.
while $flag = 0$ **do**
    (Start iteration n.)
    Step 1: Solve the master problem (MS-n) using the cutting-plane oracle from [12]
    to get an optimal solution $x^{(n)}$ and optimal objective denoted as $L^{(n)}$.
    Let $x^{(n)} = (x_Z^{(n)}, x_R^{(n)})$, where $x_Z^{(n)} \in \mathbb{Z}^{l_1}$ and $x_R^{(n)} \in \mathbb{R}^{l_2}$.
    if $x^{(n)} \in \mathcal{C}$ **then**
        Stop and return $x^{(n)}$ as an optimal solution of (MICP);
    else
        Continue.
    end if
    Step 2: Solve (7) to get the projection point $z^{(n)}$ of $x^{(n)}$ on $\mathcal{C}$.
    Add the separation cut $(x^{(n)} - z^{(n)})^\top x \le (x^{(n)} - z^{(n)})^\top z^{(n)}$ to (MS-n).
    Step 3: Solve the continuous convex program:

$$\min\ c^\top x \quad \text{s.t. } Ax \le b,\ x_Z = x_Z^{(n)}, \quad x \in \mathcal{C}. \tag{13}$$

    if (13) is feasible **then**
        Let $U^{(n)}$ be the optimal value of (13).
        if $L^{(n)} = U^{(n)}$ **then**
            Stop and return an optimal solution of (13) as an optimal solution of (MICP).
        end if
        if (13) has a solution $\bar{x}^{(n)} \in \mathrm{bd}(\mathcal{C})$. **then**
            Generate a finite number of supporting valid inequalities as illustrated in (11)
            using the oracle provided in Proposition 2.1:

$$C^{(n)} x \le d^{(n)}. \tag{14}$$

            Add $C^{(n)} x \le d^{(n)}$ to (MS-n).
            Set $\mathcal{I}^{(n)} \leftarrow \mathcal{I}^{(n-1)} \cup \{n\}$.
        end if
    else
        Set $\mathcal{I}^{(n)} \leftarrow \mathcal{I}^{(n-1)}$.
    end if
    $n \leftarrow n + 1$.
end while
Return $x^{(n)}$.

Proof We prove by contradiction. Let x^* be the limit point of $\{x^{(n)}\}_{n=1}^{\infty}$. Suppose there exists a variable $\hat{i} \in [l_1]$ such that $x^*_{\hat{i}} = \delta$ and $\delta \neq \mathbb{Z}$. Because $\{x^{(n)}\}_{n=1}^{\infty}$ converges to x^*, we have

$$\lim_{k \to \infty} \left\| x^{(k)} - x^* \right\|_1 = 0. \tag{15}$$

Therefore, for some n we have $\left\| x^{(n)} - x^* \right\|_1 < \frac{1}{2}\hat{\delta}$, where $\hat{\delta} := \min\{\delta - \lfloor \delta \rfloor, \lceil \delta \rceil - \delta\}$. On the other hand, we have

$$\left\| x^{(n)} - x^* \right\|_1 = \sum_{i=1}^{l_1+l_2} |x_i^{(n)} - x_i^*| \ge |x_{\hat{i}}^{(n)} - x_{\hat{i}}^*| = |x_{\hat{i}}^{(n)} - \delta| \ge \frac{1}{2}\hat{\delta}, \tag{16}$$

where we use the fact that $x_{\hat{i}}^{(n)} \in \mathbb{Z}$ in the last inequality. This leads to a contradiction. Therefore, we have $x^* \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2}$. Let v be the l_1-dimensional vector corresponding to the first l_1 integral components of x^*. Again due to the convergence of sequence, there exists an index N such that for any $n \geq N$ the following holds

$$\left\|x^{(n)} - x^*\right\|_1 < \frac{1}{2}. \tag{17}$$

Then the integrality of the first l_1 integer components of x^* implies that the first l_1 integer components of $x^{(n)}$ are identical to that of x^* which is the vector v, otherwise (17) is violated. This concludes the proof.

Lemma 3.1 *Suppose Algorithm 1 does not terminate finitely, and let* $\{x^{(n)}\}_{n=0}^{\infty}$ *be the infinite sequence generated by the algorithm. Then every convergent subsequence of* $\{x^{(n)}\}_{n=0}^{\infty}$ *has its limit point in* $\mathrm{bd}(\mathcal{C})$*, where* $\mathrm{bd}(\mathcal{C})$ *is the set of boundary points of* $\mathcal{C}$*.*

Proof We prove the result by contradiction. Consider a convergent subsequence $\{x^{(n_i)}\}_{i=1}^{\infty}$. Suppose

$$\hat{x} = \lim_{i \to \infty} x^{(n_i)}. \tag{18}$$

Suppose $\hat{x} \notin \mathrm{bd}(\mathcal{C})$. There are two possibilities: (1) $\hat{x} \in \mathrm{int}(\mathcal{C})$; and (2) $\hat{x} \notin \mathcal{C}$. In Case (1), there must exist a $x^{(n_i)}$ such that $x^{(n_i)} \in \mathcal{C}$. But, such a solution must be optimum because it is obtained from an outer approximation problem and Algorithm 1 should have terminated (Condition in Line 7 of Algorithm 1 is satisfied at iteration n_i). In Case (2), we have $\mathrm{dist}(\hat{x}, \mathcal{C}) > 0$. Furthermore, we have a $\hat{i}$ and $\epsilon > 0$ satisfying

$$\begin{aligned} &\mathrm{dist}(x^{(n_i)}, \hat{x}) \leq \frac{1}{4}\mathrm{dist}(\hat{x}, \mathcal{C}) \qquad \forall i \geq \hat{i}, \\ &\mathrm{dist}(x^{(n_i)}, x^{(n_j)}) \leq \frac{1}{4}\mathrm{dist}(\hat{x}, \mathcal{C}) \qquad \forall i, j \geq \hat{i}. \end{aligned} \tag{19}$$

Note that by definition $z^{(n)}$ (in (7), Line 6 of Algorithm 1) is the (Euclidean) projection of $x^{(n)}$ onto the set $\mathcal{C}$, and the inequality

$$(x^{(n)} - z^{(n)})^\top x \leq (x^{(n)} - z^{(n)})^\top z^{(n)} \tag{20}$$

added in the algorithm is a supporting valid inequality for $\mathcal{C}$ for which $z^{(n)}$ is the support point. Let

$$f^{(n)}(x) := (x^{(n)} - z^{(n)})^\top (x - z^{(n)}), \tag{21}$$

and $\mathrm{dist}(x^{(n)}, \mathcal{S}) := \min_{z \in \mathcal{S}} \|x^{(n)} - z\|_2$. We have $f^{(n)}(x^{(n)}) = (x^{(n)} - z^{(n)})^\top (x^{(n)} - z^{(n)}) = \mathrm{dist}^2(x^{(n)}, \mathcal{C})$. For any i, j satisfying $j > i \geq \hat{i}$, the following inequalities hold

$$
\begin{aligned}
&f^{(n_i)}(x^{(n_j)}) = (x^{(n_i)} - z^{(n_i)})^\top (x^{(n_j)} - z^{(n_i)}) \\
&= (x^{(n_i)} - z^{(n_i)})^\top (x^{(n_i)} - z^{(n_i)}) + (x^{(n_i)} - z^{(n_i)})^\top (x^{(n_j)} - x^{(n_i)}) \\
&\geq \mathrm{dist}^2(x^{(n_i)}, \mathcal{C}) - \|x^{(n_i)} - z^{(n_i)}\| \cdot \|x^{(n_j)} - x^{(n_i)}\| \\
&= \mathrm{dist}^2(x^{(n_i)}, \mathcal{C}) - \mathrm{dist}(x^{(n_i)}, \mathcal{C}) \cdot \mathrm{dist}(x^{(n_j)}, x^{(n_i)}) \\
&\geq \mathrm{dist}^2(x^{(n_i)}, \mathcal{C}) - \frac{1}{4}\mathrm{dist}(x^{(n_i)}, \mathcal{C}) \cdot \mathrm{dist}(\hat{x}, \mathcal{C}) \\
&\geq \left[\mathrm{dist}(\hat{x}, \mathcal{C}) - \mathrm{dist}(\hat{x}, x^{(n_i)})\right]^2 - \frac{1}{4}\left[\mathrm{dist}(\hat{x}, \mathcal{C}) + \mathrm{dist}(\hat{x}, x^{(n_i)})\right] \cdot \mathrm{dist}(\hat{x}, \mathcal{C}) \\
&\geq \frac{9}{16}\mathrm{dist}^2(\hat{x}, \mathcal{C}) - \frac{5}{16}\mathrm{dist}^2(\hat{x}, \mathcal{C}) = \frac{1}{4}\mathrm{dist}^2(\hat{x}, \mathcal{C}).
\end{aligned}
\tag{22}
$$

The above implies that $x^{(n_j)}$ violates the separation cut $f^{(n_i)}(x) \leq 0$, which leads to a contradiction since $x^{(n_j)}$ is an optimal solution of (MS-n_j) which involves the cut $f^{(n_i)}(x) \leq 0$ based on the algorithm. This concludes the proof.

The next Lemma shows that the limit point of the infinite subsequence considered in Lemma 3.1 can be identified after a certain number of iterations by solving a convex program.

Lemma 3.2 *Suppose Algorithm 1 does not terminate finitely, and let $\{x^{(n_i)}\}_{i=1}^{\infty}$ be a convergent subsequence generated by this algorithm. Let x^* be the limit point of this subsequence. Then, for a sufficiently large n_i, x^* is an optimal solution of the problem:*

$$
\begin{aligned}
&\min_{x \in \mathcal{C}} \; c^\top x \\
&\text{s.t. } Ax \leq b, \; x_j = x_j^{(n_i)}, \quad j \in [l_1].
\end{aligned}
\tag{23}
$$

Moreover, the condition at Line 20 of Algorithm 1 is satisfied for a sufficiently large n_i.

Proof By Lemma 3.1, the limit point x^* of the subsequence is in $\mathrm{bd}(\mathcal{C})$. By Proposition 3.1, there exists an iteration index N such that $x_j^{(n_i)} = x_j^* \in \mathbb{Z}$ for all $n_i > N$ and $j \in [l_1]$. Therefore, x^* is a feasible solution of (MICP). For simplicity let $x = (x_Z, x_R)$, where x_Z denote the first l_1 (integer) variables and x_R represent the continuous variables. The convex program (23) fixes the integer component to be $x_Z^{(n_i)}$. It follows that the convex program (23) is feasible at any iteration n_i for $n_i > N$ because x^* is a feasible solution of (MICP) for $n_i > N$. For any $n_i > N$, let

$$
U^{(n_i)} = \min \; c^\top x \quad \text{s.t. } Ax \leq b, \; x_j = x_j^{(n_i)}, \; j \in [l_1], \; x \in \mathcal{C}, \tag{24}
$$

$$
U^* = \min \; c^\top x \quad \text{s.t. } Ax \leq b, \; x_j = x_j^*, \; j \in [l_1], \; x \in \mathcal{C}. \tag{25}
$$

We now show that x^* is an optimal solution of (24) and (25). If not, then assume that $\hat{x}$ is an optimal solution of (25), and it satisfies $c_R^\top \hat{x}_R < c_R^\top x_R^*$.

Obviously, $(x_Z^{(n_i)}, \hat{x}_R) = (x_Z^*, \hat{x}_R)$ is a feasible solution of (25) for sufficiently large n_i. Moreover, we have

$$c^\top(x_Z^{(n_i)}, \hat{x}_R) = c_Z^\top x_Z^{(n_i)} + c_R^\top x_R^* - c_R^\top(x_R^* - \hat{x}_R) < c_Z^\top x_Z^{(n_i)} + c_R^\top x_R^*.$$

Since $\lim_{i\to\infty} x_R^{(n_i)} = x_R^*$, the above inequality implies that $c^\top(x_Z^{(n_i)}, \hat{x}_R) < c^\top x^{(n_i)}$ for sufficiently large n_i. But this contradicts with that $x^{(n_i)}$ is an optimal solution of the master problem at iteration n_i which is a relaxation of (MICP). Therefore, we have proved that x^* is an optimal solution of (25) for any n_i that is sufficiently large. Since $x_Z^{(n_i)} = x_Z^*$ for sufficiently large n_i, x^* is also an optimal solution of (24).

Lemma 3.3 *Consider the following convex program:*

$$\begin{aligned} &\min\ c^\top x \\ &\text{s.t. } Ax \le b,\ Gx = h,\ x \in \mathcal{C}. \end{aligned} \tag{26}$$

where $\mathcal{C}$ is a general convex set in $\mathbb{R}^n$. Assume that (26) *is bounded, and it has a non-empty relative interior. Suppose x^* is an optimal solution of* (26)*. Then the optimal value of the following linear program is equal to $c^\top x^*$:*

$$\begin{aligned} &\min\ c^\top x \\ &\text{s.t. } Ax \le b,\ Gx = h,\ \alpha^\top(x - x^*) \le 0, \end{aligned} \tag{27}$$

where $\alpha \in \mathcal{N}_{\mathcal{C}}(x^*)$, $r_1 \in \mathcal{N}_A(x^*)$, $r_2 \in \mathcal{N}_G(x^*)$ are three vectors obtained by the oracle in Proposition 2.1 such that $-c = \alpha + r_1 + r_2$. The notations $\mathcal{N}_A(x^*)$, $\mathcal{N}_G(x^*)$ and $\mathcal{N}_{\mathcal{C}}(x^*)$ represent the normal cones of the sets $\{x \mid Ax \le b\}$, $\{x \mid Gx = h\}$ and $\mathcal{C}$ at x^*, respectively.

Proof Since the relative interior of the feasible set in (26) is non-empty, the normal cone $\mathcal{N}(x^*)$ can be represented as [27, Corollary 23.8.1]

$$\mathcal{N}(x^*) := \mathcal{N}_A(x^*) + \mathcal{N}_G(x^*) + \mathcal{N}_{\mathcal{C}}(x^*),$$

where the sum is the Minkowski sum. Since x^* is a minimizer of (26), we have $0 \in c + \mathcal{N}(x^*)$ [28, Theorem 1.1.1]. The oracle from Proposition 2.1 can generate $r_1 \in \mathcal{N}_A(x^*)$, $r_2 \in \mathcal{N}_G(x^*)$ and $\alpha \in \mathcal{N}_{\mathcal{C}}(x^*)$ such that $-c = r_1 + r_2 + \alpha$, where r_1, r_2 and α can be zero. Notice that the equation $-c = r_1 + r_2 + \alpha$ implies that the optimality condition of (27) is satisfied at x^*, which concludes the proof.

Remark 3.1 The supporting valid inequality $\alpha^\top(x - x^*) \le 0$ given by Lemma 3.3 is added at the solution polishing Step (Lines 22-23) of Algorithm 1. Addition of these inequalities allows us to develop a proof for solving a mixed integer conic program using a finite number of inequalities.

Theorem 3.1 *Algorithm 1 terminates with an optimal solution of* (MICP) *in finitely many iterations after adding a finite number of cutting planes.*

Proof We prove by contradiction. Assume that Algorithm 1 does not terminate in a finite number of iterations. Then let $\{x^{(n)}\}_{n=0}^{\infty}$ be the master problem solutions generated by the algorithm. Since the infinite sequence is bounded, it must contain a convergent subsequence, namely $\{x^{(n_i)}\}_{i=1}^{\infty}$. Assume that x^* is the limit point of this subsequence. By Lemma 3.1, we know that $x^* \in \mathrm{bd}(\mathcal{C})$. We focus on proving that $L^{(n_i)} = U^{(n_i)}$ for a sufficiently large n_i, and hence the algorithm should terminates (Line 17).

From Proposition 3.1, and Lemma 3.2 we have an $\hat{i}$ such that $x_Z^{(n_i)} = x_Z^* \in \mathbb{Z}^{l_1}$ for any $i \geq \hat{i}$ and we have $U^{(n_i)} = U^*$, where U^* is defined as

$$\begin{aligned} U^* = \min\ & c^\top x \\ \text{s.t. } & Ax \leq b,\ x_Z = x_Z^*,\ x \in \mathcal{C}. \end{aligned} \tag{28}$$

From Algorithm 1, $x^{(n_i)}$ and $L^{(n_i)}$ are an optimal solution and the optimal value of the following mixed integer linear program:

$$\begin{aligned} \min\ & c^\top x \\ \text{s.t. } & Ax \leq b, \\ & (x^{(k)} - z^{(k)})^\top x \leq (x^{(k)} - z^{(k)})^\top z^{(k)} \quad \forall k \in [n_i - 1], \\ & C^{(k)} x \leq d^{(k)}, \quad \forall k \in \mathcal{I}^{(n_i - 1)}, \quad x \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2}. \end{aligned} \tag{29}$$

Note that the constraints $C^{(k)}x \leq d^{(k)}$ for all $k \in \mathcal{I}^{(n_i-1)}$ are the supporting valid inequalities generated at Step 3 when $\bar{x}^{(k)} \in \mathrm{bd}(\mathcal{C})$ and $c \in \mathcal{N}_{S^{(k)}}(\bar{x}^{(k)})$ (Algorithm 1, Line 22). These supporting valid inequalities pass through the point $\bar{x}^{(k)} \in \mathrm{bd}(\mathcal{C})$, which is an optimal solution of (28). Therefore, the following relation holds:

$$\begin{aligned} L^{(n_i)} &= \min\ c^\top x \text{ s.t. } \begin{cases} Ax \leq b, \quad x \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2}, \\ (x^{(k)} - z^{(k)})^\top x \leq (x^{(k)} - z^{(k)})^\top z^{(k)} \quad \forall k \in [n_i - 1], \\ C^{(k)} x \leq d^{(k)} \quad \forall k \in \mathcal{I}^{(n_i-1)} \end{cases} \\ &= \min\ c^\top x \text{ s.t. } \begin{cases} Ax \leq b, \quad x_Z = x_Z^*, \quad x \in \mathbb{R}^{l_1 + l_2}, \\ (x^{(k)} - z^{(k)})^\top x \leq (x^{(k)} - z^{(k)})^\top z^{(k)} \quad \forall k \in [n_i - 1], \\ C^{(k)} x \leq d^{(k)} \quad \forall k \in \mathcal{I}^{(n_i-1)} \end{cases} \\ &\geq \min\ c^\top x \text{ s.t. } \begin{cases} Ax \leq b, \quad x_Z = x_Z^*, \quad x \in \mathbb{R}^{l_1 + l_2}, \\ C^{(k)} x \leq d^{(k)} \qquad \forall k \in \mathcal{I}^{(n_i-1)} \end{cases} \\ &= \min\ c^\top x \text{ s.t. } \Big\{ Ax \leq b, \quad x_Z = x_Z^*, \quad x \in \mathbb{R}^{l_1 + l_2}, \quad x \in \mathcal{C}, \\ &= U^* = U^{(n_i)}, \end{aligned}$$

where we use Lemma 3.3 to replace $C^{(k)}x \leq d^{(k)}$ with $x \in \mathcal{C}$ in the third equality in the above analysis. This concludes the proof.

4 A finitely convergent cutting-plane decomposition algorithm for solving (JMICP)

We now focus on developing a finitely convergent cutting-plane decomposition algorithm for (JMICP). The general idea of this algorithm is as follows: The problem (JMICP) is decomposed into a first-stage master problem in the space of $[x,\eta]$ and a second-stage in the space of $[y,\eta]$, where η is an auxiliary variable which helps reformulating the nonlinear convex objective $g_0(x,y)$ into a constraint. In particular, the master problem at the beginning of iteration m^{th} is formulated as the following mixed 0-1 linear program:

$$\begin{aligned} \min_{x\in\mathcal{X}\cap\{0,1\}^{l_1}} \quad & \eta \\ \text{s.t.} \quad & \eta \geq a^k x + b^k \quad \forall k\in[m-1], \\ & R^{m-1}x + S^{m-1} \leq 0, \end{aligned} \tag{30}$$

where $\eta \geq a^k x + b^k$ is a valid inequality added to the first-stage at a previous iteration k, $R^{m-1}x + S^{m-1} \leq 0$ are convexification cutting planes generated before iteration m. The details on generation of these inequalities are given later. The algorithm calls the cutting-plane oracle from [7] to solve the master problem and obtain the first-stage solution x^m. Upon termination of the cutting-plane oracle, the convexification cuts are updated to be $R^m x + S^m \leq 0$.

At the second stage of iteration m, one needs to solve the following problem to optimality:

$$\begin{aligned} \min_{y\in\mathcal{Y}\cap(\mathbb{Z}^{l_2}\times\mathbb{R}^{l_3})} \quad & \eta \\ \text{s.t} \ & g_0(x^m,y) \leq \eta, \\ & g_i(x^m,y) \leq 0 \quad \forall i\in\mathcal{I}. \end{aligned} \tag{31}$$

We will develop a generalized parametric cutting-plane algorithm which shares the same spirit as the cutting-plane algorithm developed in Section 2 to solve (31), such that upon termination the parametric cutting-plane algorithm generates the following linear relaxation problem of (31):

$$\begin{aligned} \min_{\eta\in\mathbb{R},\ y\in\mathbb{R}^{l_1}\times\mathbb{R}^{l_2}} \quad & \eta \\ \text{s.t} \ & C^m x^m + D^m y + E^m \eta \leq F^m, \end{aligned} \tag{32}$$

where the matrices satisfy the following conditions:

- There exists a common optimal solution $[y^m,\eta^m]$ of (31) and (32);
- The linear inequalities $C^m x + D^m y + E^m \eta \leq F^m$ are valid for the original feasible set $\{[x,y,\eta] : \ g_0(x,y) \leq \eta,\ g_i(x,y) \leq 0\ \forall i\in\mathcal{I},\ x\in\mathcal{X}\cap\{0,1\}^{l_1},\ y\in\mathcal{Y}\cap(\mathbb{Z}^{l_2}\times\mathbb{R}^{l_3})\}$ of (JMICP).

Let $\lambda^m \geq 0$ be the optimal dual vector corresponding to the constraints in (32). We can then generate the following Bender's cut using the strong duality:

$$\eta \geq \lambda^{m\top} C^m x - \lambda^{m\top} F^m. \tag{33}$$

The above inequality will serve as an additional inequality added to the master problem (30) at the end of iteration m, i.e., $\eta \geq a^{m\top} x + b^m$ with $a^m = C^{m\top} \lambda^m$ and $b^m = -\lambda^{m\top} F^m$.

The core of the decomposition algorithm is to generate the linear relaxation (32) of (31) with the desired properties. In section 4.1, we present a parametric cutting-plane algorithm to obtain the linear relaxation, and in Section 4.2 we provide the decomposition algorithm based on the parameter cut generation. The convergence properties are analyzed.

4.1 The parametric cutting-plane algorithm

We develop a generalized parametric cutting-plane algorithm for solving a general parametric mixed-integer convex program in the form

$$\begin{aligned} \min_{y \in \mathcal{Y} \cap (\mathbb{Z}^{l_2} \times \mathbb{R}^{l_3})} \quad & h^\top y \\ \text{s.t } & g_i(x_0, y) \leq 0 \quad \forall i \in \mathcal{I}, \end{aligned} \tag{34}$$

where $x_0 \in \mathcal{X} \cap \{0,1\}^{l_1}$ is the problem parameter which is fixed when solving (34) but it is subject to change, and $\mathcal{X}$ is a polyhedral set. Notice that (31) is a special case of (34) after incorporating η as a component of y.

This algorithm is established based on the parametric cutting-plane oracle developed in [29] for solving the following parametric mixed-integer linear program to optimality:

$$\begin{aligned} \min_{y \in \mathbb{Z}^{l_2} \times \mathbb{R}^{l_3}} \quad & q^\top y \\ \text{s.t. } & W x_0 + T y \geq r, \end{aligned} \tag{35}$$

with $x_0 \in \mathcal{X} \cap \{0,1\}^{l_1}$ be the problem parameter. Assume that for any $x \in \mathcal{X} \cap \{0,1\}^{l_1}$, the problem is feasible in y. Once (35) is solved to optimality by the oracle, the oracle generates an equivalent linear program as follows:

$$\begin{aligned} \min_{y \in \mathbb{R}^{l_2} \times \mathbb{R}^{l_3}} \quad & q^\top y \\ \text{s.t. } & W' x_0 + T' y \geq r', \end{aligned} \tag{36}$$

such that the two problems have a shared optimal solution and the inequalities $W'x + T'y \geq r'$ which have incorporated all cuts generated by the oracle are valid for all $x \in \mathcal{X} \cap \{0,1\}^{l_1}$.

The generalized parametric cutting-plane algorithm works as follows: At the beginning of the n^{th} main iteration of the algorithm, we encounter a master problem in the form

$$\begin{aligned} \min \ & h^\top y \\ \text{s.t. } & a^{(k)\top} x_0 + b^{(k)\top} y + c^{(k)} \leq 0 \quad \forall k \in [n-1] \\ & C^{(k)} x_0 + D^{(k)} y \leq d^{(k)} \quad \forall k \in \mathcal{I}^{(n-1)}, \end{aligned} \tag{PMS-n}$$

where $a^{(k)\top}x_0 + b^{(k)\top}y + c^{(k)} \leq 0$ is the cutting plane generated at iteration k induced by a projection problem, and $C^{(k)}x_0 + D^{(k)}y \leq d^{(k)}$ are outer-approximation cuts generated similarly as described in Section 2. We now highlight the differences of generating the projection cuts and the outer-approximation cuts for the parametric problem as compared with the general MICP in Section 2.

Suppose $y^{(n)}$ is an optimal solution of (PMS-n). The projection problem that leads to $a^{(n)\top}x_0 + b^{(n)\top}y + c^{(n)} \leq 0$ is formulated as

$$\begin{aligned} &\min\ \|x - x_0\|^2 + \|y - y^{(n)}\|^2 \\ &\text{s.t. } g_i(x,y) \leq 0 \quad \forall i \in \mathcal{I}, \qquad y \in \mathcal{Y}. \end{aligned} \tag{37}$$

Let $[x^*, y^*]$ be the optimal solution of the projection problem. Then the coefficients of the inequality are determined by the hyperplane passing through $[x^*, y^*]$ with the norm vector $[x_0 - x^*, y^{(n)} - y^*]$. Specifically, we have $a^{(n)} = x_0 - x^*$, $b^{(k)} = y^{(n)} - y^*$ and $c^{(n)} = -(x_0 - x^*)^\top x^* - (y^{(n)} - y^*)^\top y^*$. Note that the general form of the projection cut should be $a^{(n)\top}x + b^{(n)\top}y + c^{(n)} \leq 0$. We set x to be x_0 in (PMS-n).

The supporting valid inequalities at iteration n are generated based on following similar procedures as described in Section 2. First we solve a convex optimization problem in which all integer variables are fixed:

$$\begin{aligned} &\min\ h^\top y \\ &\text{s.t } g_i(x_0, y) \leq 0 \quad \forall i \in \mathcal{I}, \\ &\quad y_Z = y_Z^{(n)},\ y \in \mathcal{Y}, \end{aligned} \tag{38}$$

where y_Z represent the integer variables in y. Let y^* be an optimal solution of (38). The linearity of the objective implies that the point $[x_0, y^*]$ must be on the boundary of the set

$$\mathcal{V} = \{[x,y]:\ g_i(x,y) \leq 0\ \forall i \in \mathcal{I},\ y \in \mathcal{Y}\}. \tag{39}$$

It is shown in Proposition 2.1 and Lemma 4.1 that one can obtain $r_i \in \text{cone}(\partial_y g_i(x_0, y))$ for every $i \in \mathcal{I}_0$ where $\mathcal{I}_0$ is the index set of active constraints at $[x_0, y^*]$, and $r' \in \mathcal{N}_{\mathcal{Y}}(y^*)$, such that the following inequalities can be generated to ensure that (45) holds when setting $x = x_0$ (see Remark 4.1):

$$r_{x,i}^\top (x - x_0) + r_i^\top (y - y^*) \leq 0 \quad \forall i \in \mathcal{I}_0, \tag{40}$$

$$r'^\top (y - y^*) \leq 0, \tag{41}$$

where $r_{x,i}$ is an arbitrary element chosen from $\partial_x g_i(x_0, y^*)$ for every $i \in \mathcal{I}_0$. The above two sets of supporting inequalities are represented in a generic form

$$C^{(n)}x + D^{(n)}y \leq d^{(n)} \tag{42}$$

in the algorithm while setting $x = x_0$, and they are added into (PMS-n) at the end of iteration n. It is shown in Theorem 4.1 that the inequalities (42) are valid for all $[x,y]$ from the set $\mathcal{S} = \{[x,y]:\ g_i(x,y) \leq 0\ \forall i \in \mathcal{I},\ y \in \mathcal{Y}\}$, which satisfies the conditions given between (32) and (33) for generating a valid Bender's cut for the first-stage problem discussed at the beginning of Section 4. This is discussed in Section 4.2. The pseudo code of the generalized parametric cutting-plane algorithm for solving (PMS-n) is given in Algorithm 2.

Lemma 4.1 *Let y^* be an optimal solution of* (38). *If the point $[x_0, y^*] \in bd(\mathcal{Y})$ then the following relation holds:*

$$h \in \sum_{i \in \mathcal{I}_0} cone(\{\partial_y g_i(x_0, y^*)\}) + cone([1_Z, 0_R]) + \mathcal{N}_{\mathcal{Y}}(y^*), \tag{43}$$

where $\mathcal{I}_0 \subseteq \mathcal{I}$ is the index set of active constraints in $\{g_i : i \in \mathcal{I}\}$ at the point $[x_0, y^*]$, $[1_Z, 0_R]$ is the vector that has all ones at the integer components and all zeros at other components of the vector y, and $\mathcal{N}_{\mathcal{Y}}(y^*)$ is the normal cone of $\mathcal{Y}$ at y^*. The oracle provided in Proposition 2.1 can generate $r_i \in cone(\{\partial_y g_i(x_0, y^*)\})$ $\forall i \in \mathcal{I}_0$, a number $\alpha \geq 0$, and $r' \in \mathcal{N}_{\mathcal{Y}}(y^*)$ such that

$$h = \sum_{i \in \mathcal{I}_0} r_i + \alpha[1_Z, 0_R] + r', \tag{44}$$

where the addition follows the Minkowski rule. Consequently, the following relation on the optimal objective holds:

$$\begin{aligned} &\min\ h^\top y \quad \text{s.t. } g_i(x_0, y) \leq 0\ \forall i \in \mathcal{I},\ y_Z = y_Z^{(n)},\ y \in \mathcal{Y} \\ &= \min\ h^\top y \quad \text{s.t. } \begin{cases} r_i^\top (y - y^*) \leq 0 \quad \forall i \in \mathcal{I}_0 \\ y_Z = y_Z^{(n)}, \\ r'^\top (y - y^*) \leq 0. \end{cases} \end{aligned} \tag{45}$$

Proof Note that (43) is exactly the optimality condition of the problem (38) at $y = y^*$, which should obviously hold. To show that (45) holds, we first observe that $y = y^*$ is a feasible solution to the linear program on the right side of (45). Furthermore, the optimality condition of the linear program is exactly the same as (44), which is clearly satisfied at $y = y^*$. This concludes the proof.

Remark 4.1 The linear constraint $r_i^\top (y - y^*) \leq 0$ for $r_i \in \text{cone}(\{\partial_y g_i(x_0, y^*)\})$ and $i \in \mathcal{I}_0$ of the linear program on the right side of (45) can be alternatively written as $r_x^\top (x - x_0) + r_i^\top (y - y^*) \leq 0$ for some $r_x \in \partial_x g_i(x_0, y)$, and then set $x = x_0$. Notice that the later inequalities in the $[x, y]$-space are valid for the set $\mathcal{V} = \{[x, y] :\ g_i(x, y) \leq 0\ \forall i \in \mathcal{I},\ y \in \mathcal{Y}\}$ guaranteed by Assumption 2. The later representation is used to generate a Bender's cut for the first-stage master problem.

Theorem 4.1 *Suppose Assumptions 2 and 3 hold. The generalized parametric cutting-plane algorithm (Algorithm 2) can solve the parametric mixed-integer convex program* (34) *to optimality in a finite number of iterations. All inequalities $a^{(k)\top} x + b^{(k)\top} y + c^{(k)} \leq 0$ and $C^{(k)} x + D^{(k)} y \leq d^{(k)}$ added to the master problem* (PMS-n) *are valid for points in the set $\mathcal{S} = \{[x, y] :\ g_i(x, y) \leq 0\ \forall i \in \mathcal{I},\ y \in \mathcal{Y}\}$.*

Proof First, we notice that all cuts added the master problem (PMS-n) are used to build an outer-approximation of the set $\mathcal{S}$, which can be seen from how these cuts are generated. Therefore, they are valid for all points in $\mathcal{S}$. It remains to show that the algorithm converges to an optimal solution of (34) in a finite number of iterations. If not, there exists a convergent sub-sequence denoted as $\{[x_0, y^{(n_k)}]\}_{k=1}^{\infty}$ which converges to a point $[x_0, y^*]$. Similarly as in the proof of Theorem 3.1, one can show that $[x_0, y^*]$ is an optimal solution of (PMS-n), $y_Z^{n_k} = y_Z^*$ for any sufficiently large n_k, and also $[x_0, y^*]$ must be a point in the sub-sequence $\{[x_0, y^{(n_k)}]\}_{k=1}^{\infty}$ due to (45). This indicates that the algorithm must terminate in a finite number of iterations.

We now discuss the need of Assumption 2 in generating a valid parametric cut. Suppose $[x_0, y^*]$ is on the boundary of the set $\mathcal{V}$ defined in (39). For simplicity, let us assume that $g_{i^*}(x, y) \leq 0$ is the only active constraint at $[x_0, y^*]$, i.e., $g_{i^*}(x_0, y^*) = 0$, $g_i(x_0, y^*) < 0$ for all $i \in \mathcal{I} \setminus \{i^*\}$ and $y^* \in int(\mathcal{Y})$. The necessary and sufficient condition for an inequality

$$a^\top (x - x_0) + b^\top (y - y^*) \leq 0 \tag{46}$$

to be valid for all points in the set $\mathcal{V}$ is $[a, b] \in \mathcal{N}_{\mathcal{V}}(x_0, y^*)$ (the normal cone of $\mathcal{V}$ at $[x_0, y^*]$), which is equivalent to $[a, b] \in \partial g_{i^*}(x_0, y^*)$ in this case. When we fix x_0 and solve (38) to get y^*, the optimality condition is written as

$$h \in \text{cone}(\partial_y g_{i^*}(x_0, y^*)) + \text{cone}([1_Z, 0_R]).$$

To get an equivalent linear program of (38), we generate a $b \in \partial_y g_{i^*}(x_0, y^*)$ and a number $\mu \geq 0$ using the oracle from Proposition 2.1 such that

$$h = b + \mu[1_Z, 0_R].$$

To lift a valid inequality generated in the space of y (fixing $x = x_0$) to the space of $[x, y]$ and maintaining its validity, requires existence of an a such that $[a, b] \in \partial g_{i^*}(x_0, y^*)$. Since in general $\partial f(x, y) \neq \partial_x f(x, y) \times \partial_y f(x, y)$, the existence of a is not guaranteed in the general case. Assumption 2 makes lifting of inequality from the y-space to $[x, y]$-space achievable.

4.2 A decomposition cutting-plane algorithm for (JMICP)

We now focus on developing a cutting-plane decomposition algorithm for solving (JMICP) to optimality. The master problem (30) can be progressively updated by adding Bender's cuts generated from solving the second-stage problem (31). Specifically, Algorithm 2 is applied to solve (31) to optimality, and when the algorithm terminates, it generates a mixed-integer linear program (MILP) that has a same optimal solution as (31). The MILP is solved to optimality with the cutting-plane oracle from [12], and the oracle generates a linear program whose optimal solution is same as that of the MILP. Suppose the linear program at the main iteration m is denoted as (32). Then the Bender's

Algorithm 2 An algorithm for solving a parametric mixed-integer convex program.

Set $n \leftarrow 1$, $flag \leftarrow 0$ and $\mathcal{I}^{(0)} \leftarrow \emptyset$.
while $flag = 0$ **do**
  (Start iteration n.)
  Step 1: Solve the master problem (PMS-n) using the cutting-plane oracle from [12]
  to get an optimal solution $y^{(n)}$ and optimal objective denoted as $L^{(n)}$.
  Let $y^{(n)} = (y_Z^{(n)}, y_R^{(n)})$, where $y_Z^{(n)} \in \mathbb{Z}^{l_2}$ and $y_R^{(n)} \in \mathbb{R}^{l_3}$.
  if $[x_0, y^{(n)}] \in \mathcal{S}$ **then**
    Stop and return $y^{(n)}$ as an optimal solution of (MICP);
  else
    Continue.
  end if
  Step 2: Solve (37) to get the projection point $[x^*, y^*]$ of $[x_0, y^{(n)}]$ on $\mathcal{S}$.
  Add the separation (projection) cut $a^{(k)\top} x_0 + b^{(k)\top} y + c^{(k)} \leq 0$ to (PMS-n).
  Step 3: Solve the continuous convex program (38).
  if (38) is feasible **then**
    Let $U^{(n)}$ be the optimal value of (38).
    if $L^{(n)} = U^{(n)}$ **then**
      Stop and return an optimal solution of (38) as an optimal solution of (PMS-n).
    end if
    if (38) has a solution y^* such that $[x_0, y^*] \in \mathrm{bd}(\mathcal{V})$ defined in (39). **then**
      Generate a finite number of supporting valid inequalities as defined in (42):

$$C^{(n)} x_0 + D^{(n)} y \leq d^{(n)} \tag{47}$$

      and add them to (PMS-n).
      Set $\mathcal{I}^{(n)} \leftarrow \mathcal{I}^{(n-1)} \cup \{n\}$.
    end if
  else
    Set $\mathcal{I}^{(n)} \leftarrow \mathcal{I}^{(n-1)}$.
  end if
  $n \leftarrow n + 1$.
end while
Return $y^{(n)}$.

cut (33) is generated and added to the master problem for the main iteration $m+1$. The pseudo code of the cutting-plane decomposition algorithm is given in Algorithm 3.

Theorem 4.2 *Suppose Assumptions 1, 2 and 3 hold. The decomposition cutting-plane algorithm (Algorithm 3) returns an optimal solution to* (JMICP) *in a finite number of iterations.*

Proof We first show that the parametric cut generated for the first-stage problem at every iteration is valid. It suffices to show that for any feasible solution $[x, y]$ of (JMICP), the following inequality holds:

$$g_0(x, y) \geq a^m x + b^m \quad \forall m, \tag{48}$$

where $\eta \geq a^m x + b^m$ is the Bender's cut generated at the main iteration m. Recall that when the generalized parametric cutting-plane algorithm is applied to solve the second-stage problem (31), it generates a master problem, and updates the master problem at every iteration by adding convexification cuts. By Theorem 4.1, the convexification cuts added to the master problem are valid for the set $\mathcal{S} = \{[x, y, \eta] : \; g_0(x, y) \leq \eta, \; g_i(x, y) \leq 0 \; \forall i \in \mathcal{I}, \; y \in \mathcal{Y}\}$. It implies that the linear program obtained upon termination of Algorithm 2 has the form

$$\min \eta \quad \text{s.t. } M_1 x^m + M_2 y + M_3 \eta \leq 0,$$

where M_1, M_2 and M_3 are some coefficient matrices, and the inequality system $M_1 x + M_2 y + M_3 \eta \leq 0$ forms a linear relaxation of $\mathcal{S}$. Then it follows that the Bender's cut $\eta \geq a^m x + b^m$ generated from the linear program should be valid for the set $\{[x, y, \eta] : \; g_0(x, y) \leq \eta, \; g_i(x, y) \leq 0 \; \forall i \in \mathcal{I}, \; x \in \mathcal{X} \; y \in \mathcal{Y}\}$.

For the first-stage solution x^m obtained at the main iteration m, let $\text{obj}_2^*(x^m)$ denote the optimal objective of the second-stage problem (31) at $x = x^m$. When the second-stage problem is solved to optimality by Algorithm 2, a linear program is obtained upon termination. The linear program has the same optimal solution and optimal value as the second-stage problem. Strong duality implies that $\text{obj}_2^*(x^m) = a^m x^m + b^m$. Let L^m and U^m be the value of L and U at the end of main iteration m. Suppose Algorithm 3 does not terminate. Since the feasible set $\mathcal{X} \cup \{0, 1\}^{l_1}$ is finite, there exist m and n such that $x^m = x^n$. We have

$$\text{obj}_2^*(x_m) = U^m \geq L^m \geq L^n \geq a^n x^n + b^n = \text{obj}_2^*(x_n). \tag{49}$$

Since $x^m = x^n$, we should have $L^m = U^m$ and hence the algorithm should terminate at the main iteration m.

Algorithm 3 A decomposition cutting-plane algorithm for (JMICP).

1: Set $m \leftarrow 1$, $L \leftarrow -\infty$ and $U \leftarrow \infty$.
2: **while** $L < U$ **do**
3: Solve the current first-stage problem (30) for the main iteration m using the cutting-plane oracle from [12], and get the optimal solution x^m.
4: Set $L \leftarrow$ the optimal value of (30).
5: Substitute x^m into the parametric MINLP (31), and apply Algorithm 2 to solve it.
6: Set $U \leftarrow$ the optimal value of (31) with $x = x^m$.
7: When Algorithm 2 terminates with an optimal solution to (31), a linear program (32) is established. Generate a Bender's cut (33) from the dual objective of (32), and add it to (30) as $\eta \geq a^m x + b^m$.
8: Set $m \leftarrow m + 1$.
9: **end while**
10: **return** x^m as an optimal solution of (JMICP).

---

5 A Parametric Cutting Plane Algorithm for Solving Distributionally-Robust Two-Stage Stochastic Mixed-Integer Convex Programs

The cutting-plane algorithm (Algorithm 1) can be naturally applied to solve a special class of two-stage mixed-integer convex programs with finite convergence. This application is established and analyzed in the current section. We consider a class of two-stage stochastic mixed-integer convex programs (TSS-MICP) with the following properties:

– The objectives of the first- and second-stage problems are linear functions;
– The first-stage variables are pure binary;
– The first-stage constraints can be nonlinear but convex;
– The second-stage decision variables can be mixed-integer;
– The second-stage constraints can be nonlinear but convex.

In particular, the TSS-MICP that we investigate is formulated as:

$$\begin{aligned} \min_{x\in\{0,1\}^{l_1}} \quad & c^\top x + \max_{P\in\mathcal{P}} \mathbb{E}_{\xi\sim P}[\mathcal{Q}(x,\xi)] \\ \text{s.t.} \quad & Ax \le b,\ x \in \mathcal{C}, \end{aligned} \tag{50}$$

where $\mathcal{C}$ is a convex set for the first-stage variable, ξ is a random vector having a finite support denoted as $\{\xi^\omega \,|\, \omega \in \Omega\}$ (Ω is a finite set of scenarios), P is a candidate probability distribution of ξ, $\mathcal{P}$ is the ambiguity set of all candidate probability distributions, and $\mathcal{Q}(x,\omega)$ is the recourse function at the scenario $\omega \in \Omega$. Note that since Ω is a finite set, an element in $\mathcal{P}$ can be represented as a $|\Omega|$-dimensional vector in $[0,1]^{|\Omega|}$. We further assume that the ambiguity set $\mathcal{P}$ is allowed to be any convex set on $\mathbb{R}^{|\Omega|}$. The recourse function $\mathcal{Q}(x,\xi^\omega)$ is given as

$$\begin{aligned} \mathcal{Q}(x,\xi^\omega) = \min_{y^\omega\in\mathbb{Z}^{l_2}\times\mathbb{R}^{l_3}} \quad & q^{\omega\top} y^\omega \\ \text{s.t. } \quad & g_j(s_j^{\omega\top} y^\omega + t_j^{\omega\top} x + r_j^\omega) \le 0 \qquad \forall j \in [J], \end{aligned} \tag{51}$$

where every g_j is a smooth convex function, q^ω, s_j^ω, t_j^ω and r_j^ω are scenario-based coefficients. We denote the minimization problem in (51) as $\mathrm{Sub}(x,\omega)$. We consider a typical iteration where the values of first stage variables are given as $\hat{x}$. Consider the second-stage sub-problem for scenario ω. Consider applying Algorithm 1 to solve this sub-problem. When the algorithm terminates, cuts have been added. Suppose the linear program at the termination is given as:

$$\begin{aligned} \min_{y^\omega\in\mathbb{R}^{l_2+l_3}} \quad & q^{\omega\top} y^\omega \\ \text{s.t. } \quad & Q^\omega y^\omega \ge s^\omega - R^\omega \hat{x}, \quad \text{dual } \mu^\omega \ge 0, \end{aligned} \tag{52}$$

where Q^ω, R^ω and s^ω are appropriate matrices and vectors of coefficients that represent the complete set of cutting planes as well as the original linear constraints at termination. The LP relaxation should give the same objective as (51). This property is formally stated as follows:

Lemma 5.1 *The termination criteria and Theorem 3.1 ensures that the optimal objective of* (52) *must be equal to* $\mathcal{Q}(\hat{x}, \xi^\omega)$.

We then determine a worst-case probability distribution with respect to the first-stage solution $\hat{x}$ by solving the following convex optimization problem (recall that $\mathcal{P}$ is a convex set in $\mathbb{R}^{|\Omega|}$)

$$\max_{p \in \mathcal{P}} \sum_{\omega \in \Omega} p^\omega \mathcal{Q}(\hat{x}, \xi^\omega) \tag{53}$$

Finally, we generate an aggregated Benders cut for the first-stage master problem based on the optimal values of dual variables associated with the constraints $Q^\omega y^\omega \geq s^\omega - R^\omega \hat{x}$ and the worst-case probability distribution p. Specifically, let μ^ω be the optimal value of the dual variables, we can construct the following aggregated Benders cut:

$$\eta \geq -\left(\sum_{\omega \in \Omega} p^\omega \mu^{\omega\top} R^\omega\right) x + \left(\sum_{\omega \in \Omega} p^\omega \mu^{\omega\top} s^\omega\right), \tag{54}$$

where p^ω is the probability of scenario ω obtained from solving (53). This cut is added to the first-stage master problem of (50). So at the m^{th} main iteration, the first-stage master problem of (50) is given as

$$\begin{aligned} &\min_{x \in \mathcal{C} \cap \{0,1\}^{l_1}} \quad c^\top x + \eta \\ &\text{s.t. } Ax \leq b, \\ &\eta \geq -\left(\sum_{\omega \in \Omega} p^\omega \mu^{k\omega\top} R^{k\omega}\right) x + \left(\sum_{\omega \in \Omega} p^\omega \mu^{k\omega\top} s^{k\omega}\right) \quad \forall k \in [m-1], \end{aligned} \tag{55}$$

where $\eta \geq -\left(\sum_{\omega \in \Omega} p^\omega \mu^{k\omega\top} R^{k\omega}\right) x + \left(\sum_{\omega \in \Omega} p^\omega \mu^{k\omega\top} s^{k\omega}\right)$ is the Benders cut added to the master problem at the k^{th} $(1 \leq k \leq m-1)$ iteration of the main algorithm. Note that (55) is a mixed 0-1 linear convex program which can again solved using Algorithm 1. The first-stage solution of (55) (the m^{th} iteration master problem) is then fed to each scenario sub-problem and generate the Benders cut indexed by iteration m. The above procedures present a typical iteration of an algorithm for solving (50). The cutting-plane method we propose for solving (50) to optimality is given in Algorithm 4. Its finite convergence is proved in the following theorem.

Theorem 5.1 *Let Assumption 3 hold. Algorithm 4 returns an optimal solution of* (50) *after finitely many iterations.*

Proof The proof is similar to the proof of Theorem 2.1 in [29]. Let the first-stage problem at main iteration k be denoted by Master-k. Lemma 5.1 and

Algorithm 4 A decomposition cutting-plane algorithm for solving (50).

```
Initialization: L ← −∞, U ← ∞, m ← 1.
while U − L > 0 do
    Solve the first-stage problem (55) to optimality with main iteration index being m.
    Let (η^m, x^m) be the optimal solution.
    Update the lower bound as L ← c^⊤ x^(m) + η^(m).
    Set the current best solution as x* ← x^m.
    for ω ∈ Ω: do
        Solve the second-stage problem Sub(x^m, ω) to optimality using Algorithm 1.
        When Algorithm 1 terminates, it generates a LP relaxation (52) with x̂ = x^m.
        Get the optimal values of the dual variables of (52) and denote them as μ^{mω}.
    end for
    Obtain the worst-case probability distribution p^m = {p^{mω} | ω ∈ Ω}
    by solving (53) with x̂ ← x^m.
    Generate the following aggregated Benders cut with iteration index m:
    η ≥ −(∑_{ω∈Ω} p^{mω} μ^{mω⊤} R^{mω}) x + (∑_{ω∈Ω} p^{mω} μ^{mω⊤} s^{mω}).
    Add the Benders cut to the first-stage problem (55).
    Update the upper bound as U ← min{U, c^⊤ x^m + ∑_{ω∈Ω} p^{mω} Q(x^m, ξ^ω)}.
    m ← m + 1.
end while
Return x*.
```

the strong duality of (52) imply that $\mathcal{Q}(x^k,\omega) = -\mu^{k\omega\top} R^{k\omega} x^k + \mu^{k\omega\top} s^{k\omega}$. Therefore, we have

$$G(x^k) := \max_{P\in\mathcal{P}} \mathbb{E}_P[\mathcal{Q}(x^k,\xi)] = -\left(\sum_{\omega\in\Omega} p^{k\omega}\mu^{k\omega\top}R^{k\omega}\right) x^k + \left(\sum_{\omega\in\Omega} p^{k\omega}\mu^{k\omega\top}s^{k\omega}\right). \tag{56}$$

Based on the mechanism of the algorithm, it is clear that if the algorithm terminates in finitely many iterations, it returns an optimal solution. We only need to show that the algorithm must terminate in finitely many iterations. Assume this property does not hold. Then it must generate an infinite sequence of first-stage solutions $\{x^k\}_{k=1}^{\infty}$. We must have k_1 and k_2 so that $x^{k_1} = x^{k_2}$, with $k_1 < k_2$. At the end of iteration k_1 the upper bound U^{k_1} satisfies

$$U^{k_1} = c^\top x^{k_1} + \sum_{\omega\in\Omega} p^{k_1\omega} Q(x^{k_1},\xi^\omega) = c^\top y^{k_1} + G(x^{k_1}), \tag{57}$$

where (56) is used to obtain the last equation. The optimal value of Master-k_2 gives a lower bound $L^{k_2} = c^\top x^{k_2} + \eta^{k_2}$. Since $k_2 > k_1$, Master-k_2 has the following constraint:

$$\eta \geq -\left(\sum_{\omega\in\Omega} p^{k_1\omega}\mu^{k_1\omega\top}R^{k_1\omega}\right) x + \left(\sum_{\omega\in\Omega} p^{k_1\omega}\mu^{k_1\omega\top}s^{k_1\omega}\right). \tag{58}$$

Therefore, we conclude that

$$\begin{aligned} L^{k_2} &= c^\top x^{k_2} + \eta^{k_2} \\ &\geq c^\top x^{k_2} - \left(\sum_{\omega\in\Omega} p^{k_1\omega}\mu^{k_1\omega\top}R^{k_1\omega}\right)x^{k_2} + \left(\sum_{\omega\in\Omega} p^{k_1\omega}\mu^{k_1\omega\top}s^{k_1\omega}\right) \\ &= c^\top x^{k_1} - \left(\sum_{\omega\in\Omega} p^{k_1\omega}\mu^{k_1\omega\top}R^{k_1\omega}\right)x^{k_1} + \left(\sum_{\omega\in\Omega} p^{k_1\omega}\mu^{k_1\omega\top}s^{k_1\omega}\right) \\ &= c^\top x^{k_1} + G(y^{k_1}) = U^{k_1} = U^{k_2}, \end{aligned} \tag{59}$$

where we use the fact that $x^{k_1} = x^{k_2}$, and inequalities (57)–(58) to obtain (59). Hence, we have no optimality gap at solution x^{k_1}, and the algorithm should have terminated at or before iteration k_2.

6 A Numerical Example of the Cutting-Plane Algorithm

Let us consider the following TSS-MICP:

$$\begin{aligned} &\min_{x_1,x_2\in\{0,1\}} \; x_1 + 2x_2 + \mathbb{E}_\xi[\mathcal{Q}(x,\xi^\omega)] \\ &\text{s.t. } 3x_1 + x_2 \geq 2, \end{aligned} \tag{60}$$

where there are only two scenarios $\Omega = \{\omega_1, \omega_2\}$ with equal probability. The recourse function at the two scenarios are given as

$$\begin{aligned} \mathcal{Q}(x,\xi^{\omega_1}) = &\min_{y_{11},y_{12}\in\mathbb{Z}_+} \; 0.5y_{11} + y_{12} \\ &\text{s.t. } -2y_{11} - [\log(1+e)]y_{12} + \log(1+e^{y_{11}+y_{12}}) \leq x_1 + x_2 - 1. \end{aligned} \tag{61}$$

$$\begin{aligned} \mathcal{Q}(x,\xi^{\omega_2}) = &\min_{y_{21},y_{22}\in\mathbb{Z}_+} \; y_{21} + y_{22} \\ &\text{s.t. } -[\log(1+e)](y_{21}+y_{22}) + \log(1+e^{y_{21}+y_{22}}) \leq x_1 + x_2 - 1. \end{aligned} \tag{62}$$

The master problem at Iteration 1 is simply

$$\min \; x_1 + 2x_2 + \eta \quad \text{s.t. } 3x_1 + x_2 \geq 2, \; \eta \geq 0, \; x_1, x_2 \in \{0,1\}.$$

The relaxation problem yields the fractional solution $[x_1, x_2] = [2/3, 0]$. We further add a cut $x_1 \geq 1$ and resolve the master problem, which gives the optimal solution $[x_1, x_2] = [1, 0]$. Substituting the first-stage solution $[x_1, x_2] = [1, 0]$ into (61) and solving the continuous relaxation sub-problem yields a fractional solution $[y_{11}, y_{12}] = [0.5, 0.48]$ which is on the curve $-2y_{11} - [\log(1+e)]y_{12} + \log(1+e^{y_{11}+y_{12}}) = 0$. We add the tangent cut $1.272y_{11} + 0.586y_{12} \geq 0.918$ induced by this fraction solution to the relaxation sub-problem. Then we solve the following relaxed MILP:

$$\min \; 0.5y_{11} + y_{12} \quad \text{s.t. } -1.272y_{11} - 0.586y_{12} + 0.918 \leq 0, \; y_{11}, y_{12} \in \mathbb{Z}_+,$$

by adding a cut $y_{11} + y_{12} \geq 1$ to the linear relaxation of the above MILP, which leads to the integral solution $[y_{11}, y_{12}] = [1, 0]$ of (61) for the given first-stage value $[x_1, x_2] = [1, 0]$. Similarly, substituting $[x_1, x_2] = [1, 0]$ into (62), and solving the continuous relaxation gives the solution $[y_{21}, y_{22}] = [0.5, 0.5]$. We add a tangent cut $y_{21} + y_{22} \geq 1$ induced by this fractional solution to the relaxation sub-problem. Then we solve the following relaxed MILP:

$$\min\ y_{11} + y_{12} \quad \text{s.t.}\ - y_{11} - y_{12} + 1 \leq 0,\ y_{11}, y_{12} \in \mathbb{Z}_+,$$

which leads to an integral optimal solution $[y_{21}, y_{22}] = [1, 0]$.

Notice that the eventual linear program that leads to the integral solution $[y_{11}, y_{12}] = [1, 0]$ of (61) is:

$$\min\ 0.5y_{11} + y_{12} \quad \text{s.t.}\ - y_{11} - y_{12} + 1 \leq x_1 + x_2 - 1,\ y_{11}, y_{12} \geq 0.$$

The Bender's cut associated with this scenario problem is $\eta^{\omega_1} \geq 1 - 0.5x_1 - 0.5x_2$. Similarly, the Bender's cut associated with the scenario ω_2 problem is $\eta^{\omega_2} \geq 2 - x_1 - x_2$. The aggregated Bender's cut is $\eta \geq 1.5 - 0.75x_1 - x_2$. Adding this aggregated Bender's cut to the master problem gives:

$$\min\ x_1 + 2x_2 + \eta \quad \text{s.t.}\ 3x_1 + x_2 \geq 2,\ \eta \geq 1.5 - 0.75x_1 - x_2,\ x_1, x_2 \in \{0, 1\},\ \eta \geq 0.$$

Solving the above updated master problem gives the first-stage solution $[x_1, x_2] = [1, 0]$, which is the same as the previous iteration. This indicates that the optimal solution is $[x_1^*, x_2^*] = [1, 0]$.

References

1. Padberg, M., Rinaldi, G.: A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Review **33**(1), 60–100 (1991)
2. Mitchell, J.E.: Branch-and-cut algorithms for combinatorial optimization problems. In: P.M. Pardalos, M.G.C. Resende (eds.) Handbook of applied optimization, 1, pp. 65–77. Oxford University Press (2002)
3. Cornuéjols, G.: Valid inequalities for mixed integer linear programs. Mathematical Programming **112**(1), 3–44 (2008)
4. Balas, E., Ceria, S., Cornuéjols, G., Natraj, N.: Gomory cuts revisited. Operations Research Letters **19**(1), 1–9 (1996)
5. Padberg, M.W., Roy, T.J.V., Wolsey, L.A.: Valid linear inequalities for fixed charge problems. Operations Research **33**(4), 842–861 (1985)
6. Roy, T.J.V., Wolsey, L.A.: Valid inequalities and separation for uncapacitated fixed charge networks. Operations Research Letters **4**(3), 105–112 (1985)
7. Balas, E., Ceria, S., Cornuéjols, G.: A lift-and-project cutting plane algorithm for mixed 0-1 programs. Mathematical Programming **58**, 295–324 (1993)
8. Balas, E.: Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. SIAM. J. on Algebraic and Discrete Methods **6**(3), 466–486 (1985)
9. Balas, E.: Disjunctive programming: properties of the convex hull of feasible points. Discrete Applied Mathematics **89**, 3–44 (1998)
10. Owen Jonathan H. Mehrotra, S.: A disjunctive cutting plane procedure for general mixed-integer linear programs. Mathematical Programming **89**(3), 437–448 (2001)
11. Owen, J.H., Mehrotra, S.: A disjunctive cutting plane procedure for general mixed-integer linear programs. Computational Optimization and Applications **20**, 159–170 (2001)

12. Chen, B., Simge Küçükyavuz, S.S.: Finite disjunctive programming characterizations for general mixed-integer linear programs. Operations Research **59**(1), 202–210 (2011)
13. Jörg, M.: k-disjunctive cuts and a finite cutting plane algorithm for general mixed integer linear programs (2007). `https://arxiv.org/pdf/0707.3945.pdf`
14. Duran, M., Grossmann, I.: An outer-approximation algorithm for a class of mixed integer nonlinear programs. Mathematical Programming **36**(3), 307–339 (1986)
15. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Mathematical Programming **103**(2), 225–249 (2005)
16. Hijazi, H., Bonami, P., Ouorou, A.: An outer-inner approximation for separable mixed-integer nonlinear programs. INFORMS Journal on Computing **26**(1), 31–44 (2014)
17. Vielma, J.P., Dunning, I., Huchette, J., Lubin, M.: Extended formulations in mixed integer conic quadratic programming. Mathematical Programming Computation **9**(3), 369–418 (2017)
18. Abhishek, K., Leyffer, S., Linderoth, J.: Filmint: An outer approximation-based solver for convex mixed-integer nonlinear programs. informs journal on computing. INFORMS Journal on Computing **22**(4), 555–567 (2010)
19. Bonami, P., Biegler, L., Conn, A., , Cornuéjols, G., Grossmann, I., Laird, C., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wachter, A.: An algorithmic framework for convex mixed integer nonlinear programs. Discrete Optimization **5**(2), 186–204 (2008)
20. Lubin, M., Yamangil, E., Bent, R., Vielma, J.P.: Polyhedral approximation in mixed-integer convex optimization. Mathematical Programming pp. 139–168 (2018)
21. Gade, D., Küçükyavuz, S., Sen, S.: Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. Mathematical Programming **144**, 39–64 (2014)
22. Zhang, M., Küçükyavuz, S.: Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs. SIAM Journal on Optimization **24**(4), 1933–1951 (2014)
23. Bansal, M., Huang, K.L., Mehrotra, S.: Tight second stage formulations in two-stage stochastic mixed integer programs. SIAM Journal on Optimization **28**, 788–819 (2018)
24. Bansal, M., Zhang, Y.: Scenario-based cuts for structured two-stage stochastic and distributionally robust p-order conic mixed integer programs. Journal of Global Optimization (2021). URL `https://doi.org/10.1007/s10898-020-00986-w`
25. Atamtürk, A., Narayanan, V.: Conic mixed-integer rounding cuts. Mathematical Programming **122**, 1–20 (2008)
26. Sen, S., Sherali, H.D.: Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. Mathematical Programming **106**, 203–223 (2006)
27. Rochafellar, R.T.: Convex Analysis. Princeton University Press (1970)
28. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization - I. Springer-Verlag (1993)
29. Luo, F., Mehrotra, S.: A decomposition method for distributionally-robust two-stage stochastic mixed-integer conic programs. Mathematical Programming (2021). `https://doi.org/10.1007/s10107-021-01641-2`