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

Fengqiao Luo · Shibshankar Dey · Sanjay Mehrotra

Received: date / Accepted: date

Abstract We present a finitely convergent cutting-plane algorithm for solving a general mixed-integer convex programs given an oracle for solving general convex programs. This method is extended to solve a family of two-stage mixed-integer convex programs using cutting planes, with applications to solving distributionally-robust two-stage stochastic mixed-integer convex programs. Since algorithms purely using cutting planes are not very practical for implementation, we combined the cut generation with a branch-and-union scheme to develop a more practical algorithm. Analysis is also given for the case where convex programming oracle provides an $\epsilon-$optimal solution. Computational results on generated test problems show the practicality of our algorithm. Specifically, results show that addition of cuts speed up solution times by nearly 10-fold on the largest test problems that are solved.

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

Shibshankar Dey
Department of Industrial Engineering and Management Science, Northwestern University
Evanston, Illinois
shibshankardey2025@u.northwestern.edu

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

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

\[
\begin{align*}
\min & \quad c^\top x + h^\top y \\
\text{s.t.} & \quad Tx + Wy = q, \\
& \quad g_i(x, y) \leq 0 \quad \forall i \in I, \\
& \quad x \in \mathcal{X} \cap \{0, 1\}^{l_1}, \ y \in \mathcal{Y} \cap (\mathbb{Z}^{l_2} \times \mathbb{R}^{l_3}),
\end{align*}
\]

(JMICP)

where \(x\) are binary variables, and \(y\) are mixed-integer variables. \(Tx + Wy = q\) represents polyhedral constraints. The functions \(g_i(x, y)\) for \(i \in I\) are convex but not necessarily differentiable, and \(\mathcal{X}\) and \(\mathcal{Y}\) are bounded polyhedral sets for \(x\) and \(y\) including simple-bound constraints. Note that the following problem can be formulated as (JMCP):

\[
\begin{align*}
\min & \quad g_0(x, y) \\
\text{s.t.} & \quad Tx + Wy = q, \\
& \quad g_i(x, y) \leq 0 \quad \forall i \in I, \\
& \quad x \in \mathcal{X} \cap \{0, 1\}^{l_1}, \ y \in \mathcal{Y} \cap (\mathbb{Z}^{l_2} \times \mathbb{R}^{l_3}),
\end{align*}
\]

where \(g_0(x, y)\) is a convex function of \((x, y)\). It can be achieved by moving the convex objective to the set of constraints with an auxiliary variable \(\eta\) as \(g_0(x, y) \leq \eta\). In this case, the new objective is to minimize \(\eta\) and we re-define a new \(\mathcal{Y}\)-space variable \(\tilde{y}\) as \(\tilde{y} := [y; \eta]\), and (1.1) set with \(c = 0, h = (0; 1)\).

An important special case of (1.1) is the following two-stage stochastic mixed-integer convex program (TSS-MICP) under finite support assumption:

\[
\begin{align*}
\min & \quad c(x) + \mathbb{E}_\xi[Q(x, \xi)] \\
\text{s.t.} & \quad Ax \leq b, \ x \in \mathcal{C}, \\
& \quad x \in \{0, 1\}^{l_1},
\end{align*}
\]

(1.2)

where \(x\) is the vector of first-stage variables that are pure binary, \(\mathcal{C}\) is a closed convex set, and \(\xi\) is a vector of random parameters that follow the joint distribution \(P\) on a finite support \(\Omega\). The recourse function \(Q(x, \xi_\omega)\) for scenario \(\omega \in \Omega\) is given as:

\[
\begin{align*}
Q(x, \xi_\omega) &= \min_{y^\omega} h^\omega^\top y^\omega \\
\text{s.t.} & \quad T^\omega x + W^\omega y^\omega = q^\omega, \\
& \quad g_i^\omega(x, y^\omega) \leq 0 \quad \forall i \in I, \\
& \quad y^\omega \in \mathbb{Z}^{l_2} \times \mathbb{R}^{l_3},
\end{align*}
\]

(1.3)

where \(g_i^\omega\) are convex functions. Under finiteness assumption of support set \(\Omega\), (1.2) and (1.3) admit an extended reformulation in terms of variables \(x\) and \(y^\omega\) for
\( \omega \in \Omega: \)
\[
\min_{\omega \in \Omega} c(x) + \sum_{\omega \in \Omega} p^\omega h^{\omega^T} y^\omega
\]
\[
\text{s.t. } T^\omega x + W^\omega y^\omega = q^\omega, \\
g_i^\omega (x, y^\omega) \leq 0 \quad \forall i \in I, \\
Ax \leq b, x \in C, \\
x \in \{0, 1\}^n, y^\omega \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2},
\]
where \( p^\omega \) is the probability of scenario \( \omega \). Note that (1.4) is in the form of (1.1).

Thus (1.4) can be solved directly using an algorithm for solving mixed integer convex programs. However, a Benders’ decomposition approach is typically preferable when the number of scenarios is large.

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 as well as a linearize-branch-and-union decomposition algorithm for solving (JMIPC) 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 \( x \in X \cap \{0, 1\}^{l_1} \), (JMIPC) is feasible in \( y \).

**Assumption 2** Interior of the convex set \( S := \{[x, y] : g_i(x, y) \leq 0 \forall i \in I, y \in \mathcal{Y}\} \) is non-empty. Additionally, for any fixed \( x \in X \cap \{0, 1\}^{l_1} \), (JMIPC) has non-empty relative interior in \( y \).

**Assumption 3** For any \( (x_0, y_0) \in \mathbb{R}^{l_1+l_2+l_3} \), the function \( g_i \forall i \in I \cup \{0\} \) 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 4** We have an oracle that can solve a convex optimization problem to optimality.

Assumptions 1-4 simplify our presentation and analyses in Sections 4-6. An analysis of the algorithms for mixed-integer convex program (MICP), joint mixed-integer convex program (JMIPC) and TSS-MICP without assumption 4 is provided in Appendix B under \( \epsilon \)-optimality of the convex programming oracle. It is possible to add artificial variables to ensure Assumption 1. In particular, adding artificial variables \( z_i, i \in I \) results in the following formulation:

\[
\min c^T x + h^T y + M \sum_{i \in I} z_i + M \sum_{j \in J} (z_j^+ + z_j^-) \tag{1.5}
\]
\[
\text{s.t. } Tx + Wy = q + z^+ - z^-, \\
g_i(x, y) \leq M z_i, \quad i \in I \tag{1.6} \\
x \in X \cup \{0, 1\}^{l_1}, \quad y \in \mathcal{Y} \cup (\mathbb{Z}^{l_2} \times \mathbb{R}^{l_3}), \tag{1.7} \\
z_i \in \{0, 1\}, \quad i \in I, \quad z^+, z^- \in \mathbb{R}_+^{\lvert J \rvert} \tag{1.8}
\]
where $M$ is large, $z_i$’s and $z^+_j, z^-_j$’s are auxiliary binary and non-negative variables that ensure feasibility.

Non-empty interior of $S$ in Assumption 2 implies that Slater condition holds which is a common assumption in convex optimization literature. Assumption 3 is used when ensuring that the cutting planes generated to solve (JMIPC) in the space of $y$ for any fixed $x$ are valid and extendable to generate a Benders’ cut in the space of $x$. More detailed explanation regarding this assumption is provided at the end of Section 4.1. Note that Assumption 3 is satisfied by two important families of functions: (1) Differentiable convex functions in the $(x, y)$-space; (2) Separable convex functions in the form of $g(x, y) = \psi(x) + \phi(y)$, where $\psi$ and $\phi$ are convex. Two-stage distributionally robust stochastic convex programs also satisfy this assumption.

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 [5], mixed-integer rounding cuts, and polyhedral structure based cuts such as flow cover inequalities [6] and flow-path inequalities [7]. 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 [8], 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 [5, 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 an algorithm’s convergence to an optimal solution, cuts are added at all $\gamma$-optimal (a notion defined in [10]) vertices of the LP-relaxation master problem at every iteration. As an alternative to exploring the $\gamma$-optimal vertices, Chen et. al. [11] 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 [12]
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 [3,9].

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 [13]. 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. [14] 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 [14] is incomplete, and is not applicable for the case where the convex functions are not differentiable. The convex functions considered in [14] 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 [15,16]. Gade et al. [15] 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 [16] 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 [17], and references therein). For two-stage stochastic mixed-integer conic optimization, Bansal and Zhang [18] 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 [19] on convexifying a simple polyhedral conic mixed-integer set.

Sen and Sherali [20] 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 Benders’ 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 [29]. In [21] 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.

1.2 Contributions of this paper

This paper makes the following contributions:

– A cutting-plane algorithm is developed for solving a general mixed-integer convex program (MICP) (Section 2). It is proved that the algorithm can identify an optimal solution of 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 a 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). This decomposition algorithm is further generalized to solve a family of distributionally-robust two-stage stochastic mixed integer convex programs (Section 5).

– For practical implementation, a cutting-plane algorithm often needs to be coupled with a branch-and-bound technique. Therefore, we combined the cut generation and the branch-and-bound method to develop a linearize-branch-and-union algorithm that is more suitable for implementation (Section 6).

– The main algorithms developed in this paper were implemented and tested to solve a family of distributionally-robust two-stage stochastic mixed-integer convex programs with pure-binary first stage decision variables and mixed-integer second stage decision variables (Section 7). Computational results show that addition of cuts reduce the solution time by nearly 10-fold for problems that could be solved without adding cuts within a few days.

– We provide an error analysis of our methods for the case where the convex-optimization oracle can only solve a convex programs to a desired accuracy (Section 8).

To this end, we introduce some notations used in our analysis throughout the next sections.

**Notations:** We use $\mathcal{C}$ to represent a closed convex set. dist$(x, \mathcal{C})$ denotes the minimum distance between point $x$ and set $\mathcal{C}$. dist$(x, y)$ defines the distance between any two point $x$ and $y$. bd$(\mathcal{C})$ represents the set of boundary point of
non-empty convex set $C$. At some point $x \in \text{bd}(C)$, a normal cone is denoted with $\mathcal{N}_C(x)$. $\|\cdot\|$ denotes 2-norm unless specified with a subscript. We let $[K]$ denote the index set $\{1, 2, \ldots, K\}$ and $|K|$ be the cardinality.

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

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

$$\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b, \quad x \in C, \quad x \in \mathbb{Z}^l_1 \times \mathbb{R}^l_2,
\end{align*}$$

where $Ax = b$ are linear constraints and $C$ is a general convex set including non-negativity constraints and is assumed to have a non-empty interior. However, feasible region of (MICP) does not necessarily have any interior; it is only assumed to have non-empty relative-interior. 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 $C$ can be described by inequality constraints $c_i(x) \leq 0$ for $i \in [m]$, where $[m]$ represents the set of integers $\{1, \ldots, m\}$, and $c_i(\cdot) : \mathbb{R}^{l_1+\ell_2} \to \mathbb{R}$ are proper convex functions. In another important special case, the set $C := \bigcap_{t=1}^T K_t$, where $K_t$ are proper convex cones. For any non-empty convex set $C \subset \mathbb{R}^n$, $C \neq \mathbb{R}^n$, a point $x \in C$ is on the boundary if and only if there is a hyperplane supporting $C$ at $x$. We use $\text{bd}(\cdot)$ to denote the set of boundary points. For general convex set $S \subset \mathbb{R}^n$, at some point $x \in \text{bd}(S)$ a normal cone is denoted with $\mathcal{N}_S(x)$. We start with the following preliminary results which are used in the cutting-plane algorithm for solving (MICP) developed in this section.

**Proposition 2.1** For general convex sets $S_k \subset \mathbb{R}^n$ with $k \in [K] := \{1, 2, \ldots, K\}$, for an point $x \in \text{bd}(\bigcap_{k=1}^K S_k)$ let $\mathcal{N}_{S_k}(x)$ be the normal cone of $S_k$ at $x$. For any vector $c$ satisfying

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

where the addition follows the Minkowski rule, we can generate an oracle that can generate $K$ points $v_k$ for $k \in [K]$ (some of them can be zero) such that

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

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

$$\begin{align*}
\min & \quad \sum_{k=1}^K \|v_k\|_2 \\
\text{s.t.} & \quad -c = \sum_{k=1}^K v_k, \quad v_k \in \mathcal{N}_{S_k}(x) \quad \forall k \in [K].
\end{align*}$$
Remark 2.1 We note that the cone $\mathcal{N}_{S_k}(x)$ may not be finitely generated. Also, 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 (2.3) can be solved numerically. To give an example, assume $K = 2$ with second-order cones $S_1 = \{(x, t) : \|A_1 x\|_2 \leq b_1 t\}$ and $S_2 = \{(x, t) : \|A_2 x\|_2 \leq b_2 t\}$, where $[x, t] \in \mathbb{R}^n \times \mathbb{R}$, $c_1, c_2 > 0$, and $A_1, A_2$ are invertible. Consider the normal cone of $S_1$ and $S_2$ at $y = [0, 0]$, which are given as $\mathcal{N}_{S_1} = \{(x, t) : \| (A_1^{-1})^\top x \| \leq -t/b_1 \}$ and $\mathcal{N}_{S_2} = \{(x, t) : \| (A_2^{-1})^\top x \| \leq -t/b_2 \}$. Then the optimization problem (2.3) becomes the following second-order cone program:

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

In the above example, for simplicity, we have kept both second order cones be pointed at the origin. However, the proposition allows for these cones be pointed at different points; and thus not requiring a translation of cones not pointed at 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 $\hat{x} \in \text{bd}(S)$, i.e., it is a supporting hyperplane of $S$ at $\hat{x}$.

Proposition 2.2 Let $C$ be a general convex set in $\mathbb{R}^n$. Consider the following convex program:

$$\begin{align*}
\min & \quad c^\top x \\
\text{s.t.} & \quad A' x = b', \quad x \in C.
\end{align*}$$

Assume that (2.4) is bounded, and it has a non-empty relative interior. Let $x^*$ be an optimal solution of (2.4), and $\alpha \in \mathcal{N}_C(x^*)$, a normal cone of set $C$ at $x^*$ is $\mathcal{N}_C(x^*)$. Then the optimal value of the following linear program is equal to $c^\top x^*$:

$$\begin{align*}
\min & \quad c^\top x \\
\text{s.t.} & \quad A' x = b', \quad \alpha^\top (x - x^*) \leq 0,
\end{align*}$$

Proof Let $\mathcal{N}_A(x^*)$ represent the normal cone of the set $\{x | A' x = b'\}$. Since the relative interior of the feasible set in (2.4) is non-empty, the normal cone $\mathcal{N}(x^*)$ can be represented as $[22 \text{ Corollary 23.8.1}]$

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

where the sum is the Minkowski sum. Since $x^*$ is a minimizer of (2.4), we have $0 \in c + \mathcal{N}(x^*)$ [23 Theorem 1.1.1]. The oracle from Proposition 2.1 can
generate \( r \in \mathcal{N}_A(x^*) \) and \( \alpha \in \mathcal{N}_C(x^*) \) such that \(-c = r + \alpha\), where \( r \) and \( \alpha \) can be zero. Notice that the equation \(-c = r + \alpha\) implies that the optimality condition of (2.5) is satisfied at \( x^* \), which concludes the proof.

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:

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

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 \( C \) at iteration \( k \), and \( C^{(k)} x \leq d^{(k)} \) are outer-approximation cuts generated to approximate \( C \) 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 [11] for solving a general mixed-integer linear program 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 \( x^{(n)} \in C \), then \( x^{(n)} \) is optimal to (MICP) and the algorithm stops. Otherwise, consider the projection problem:

\[
\min_{z \in C} \|z - x^{(n)}\|_2^2
\]

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

\[
(x^{(n)} - z^{(n)})^\top x \leq (x^{(n)} - z^{(n)})^\top z^{(n)}. \tag{2.7}
\]

The algorithm first adds (2.7) to (MS-\( n \)). Then it solves the following continuous convex program:

\[
\min_{x \in \mathcal{C}} c^\top x \\
\text{s.t. } Ax = b, \\
x_j = x^{(n)}_j \quad j \in [l_1].
\]

The problem in (2.8) 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 (2.8) is infeasible; (b) Find an optimal solution \( \bar{x}^{(n)} \) of (2.8), 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 $S^{(n)} = \{ x : A'x = b', \ x \in \mathcal{C} \}$ where $A'x = b'$ represents the constraints $Ax = b$, $x_j = x_j^{(n)} \forall j \in [l_1]$ together. The algorithm runs the oracle given in Proposition 2.1 to obtain $r' \in \mathcal{N}_{S^{(n)}}(\bar{x}^{(n)})$ and $r \in \mathcal{N}_C(\bar{x}^{(n)})$ satisfying $-c = r' + r$, where $S^{(n)} = \{ x : A'x = b' \}$. In this case, the algorithm constructs the following supporting linear inequality

$$r^\top (x - \bar{x}^{(n)}) \leq 0. \quad (2.9)$$

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

$$C^{(n)} x \leq d^{(n)}. \quad (2.10)$$

By Proposition 2.2, the oracle ensures the following relation:

$$\begin{align*}
\{ \min c^\top x \mid Ax = b, \ x_j = x_j^{(n)} \forall j \in [l_1], \ x \in \mathcal{C} \} & = \{ \min c^\top x \mid Ax = b, \ x_j = x_j^{(n)} \forall j \in [l_1], \ C^{(n)} x \leq d^{(n)} \}. \quad (2.11)
\end{align*}$$

The additional cuts (2.10) are added to $\{\text{MS-}n\}$. 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 $\bar{x}^{(k)}$ 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

$$\begin{align*}
\{ \min c^\top x \mid Ax = b, \ x_j = x_j^{(k)} \forall j \in [l_1], \ x \in \mathcal{C} \}
\end{align*}$$

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 $\{ k \in [n] \mid \bar{x}^{(k)} \in \text{bd}(\mathcal{C}), -c \in \mathcal{N}_{S^{(k)}}(\bar{x}^{(k)}) \}$ 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 $v \in \mathbb{Z}^{l_3}$ 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 (MICP).

1: Set $n \leftarrow 1$, $flag \leftarrow 0$ and $I^{(0)} \leftarrow \emptyset$.
2: while $flag = 0$ do
3: (Start iteration $n$.)
4: Step 1: Solve the master problem (MS-$n$) using the cutting-plane oracle from \cite{11} to get an optimal solution $x^{(n)}$ and optimal objective denoted as $L^{(n)}$.
5: Let $x^{(n)} = (x_Z^{(n)}, x_R^{(n)})$, where $x_Z^{(n)} \in \mathbb{Z}^{l_1}$ and $x_R^{(n)} \in \mathbb{R}^{l_2}$.
6: if $x^{(n)} \in C$ then
7: Stop and return $x^{(n)}$ as an optimal solution of (MICP).
8: else
9: Continue.
10: Step 2: Solve (2.6) to get the projection point $z^{(n)}$ of $x^{(n)}$ on $C$.
11: Add the separation cut $(x^{(n)} - z^{(n)})^\top x \leq (x^{(n)} - z^{(n)})^\top z^{(n)}$ to (MS-$n$).
12: Step 3: Solve the continuous convex program:
13: \begin{equation}
\min_{x} c^\top x \quad \text{s.t.} \quad Ax = b, \ x_Z = x_Z^{(n)}, \ x \in C.
\end{equation} (2.12)
14: if (2.12) is feasible then
15: Let $U^{(n)}$ be the optimal value of (2.12).
16: if $L^{(n)} = U^{(n)}$ then
17: Stop and return an optimal solution of (MICP) as an optimal solution of (2.12).
18: end if
19: if (2.12) has an optimal solution $\bar{x}^{(n)} \in \text{bd}(C)$, then
20: Generate a supporting valid inequality as described before (2.10) using the oracle provided in Proposition 2.1:
21: \begin{equation}
C^{(n)}x \leq d^{(n)}.
\end{equation} (2.13)
22: Add $C^{(n)}x \leq d^{(n)}$ to (MS-$n$).
23: Set $I^{(n)} \leftarrow I^{(n-1)} \cup \{n\}$.
24: else
25: Set $I^{(n)} \leftarrow I^{(n-1)}$.
26: end if
27: \end{if}
28: else
29: Set $I^{(n)} \leftarrow I^{(n-1)}$.
30: end if
31: $n \leftarrow n + 1$.
32: end while
33: 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 $\delta \notin \mathbb{Z}$. Because $\{x^{(n)}\}_{n=1}^\infty$ converges to $x^*$, we have

\begin{equation}
\lim_{k \to \infty} \|x^{(k)} - x^*\|_1 = 0.
\end{equation} (3.1)
Therefore, for some \( n \) we have \( \|x^{(n)} - x^*\|_1 < \frac{1}{2} \hat{\delta} \), where \( \hat{\delta} := \min\{\delta - \lfloor \delta \rfloor, \lceil \delta \rceil - \delta\} \). On the other hand, we have \( x^{(n)} - x^* = l_1 + l_2 \sum_{i=1}^{l_1+l_2} |x_i^{(n)} - x_i^*| \geq |x_1^{(n)} - x_1^*| = |x_1^{(n)} - \delta| \geq \frac{1}{2} \hat{\delta} \), (3.2)

where we use the fact that \( x_i^{(n)} \in \mathbb{Z} \) 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

\[
\|x^{(n)} - x^*\|_1 < \frac{1}{2}.
\]

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 (3.3) 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 \( \text{bd}(C) \), where \( \text{bd}(C) \) is the set of boundary points of \( C \).

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

\[
\hat{x} = \lim_{i \to \infty} x^{(n_i)}.
\]

Suppose \( \hat{x} \notin \text{bd}(C) \). There are two possibilities: (1) \( \hat{x} \in \text{int}(C) \); and (2) \( \hat{x} \notin C \). In Case (1), there must exist a \( x^{(n_i)} \) such that \( x^{(n_i)} \in C \). 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 10 of Algorithm 1 is satisfied at iteration \( n_i \)). In Case (2), we have dist(\( \hat{x}, C \)) > 0. Furthermore, because \( x^{(n_i)} \) is convergent, we have a \( i \) satisfying

\[
\text{dist}(x^{(n_i)}, \hat{x}) \leq \frac{1}{4} \text{dist}(\hat{x}, C) \quad \forall i \geq \hat{i},
\]
\[
\text{dist}(x^{(n_i)}, x^{(n_j)}) \leq \frac{1}{4} \text{dist}(\hat{x}, C) \quad \forall i, j \geq \hat{i}.
\]

(3.5)

Note that by definition \( z^{(n)} \) (in (2.6), Line 6 of Algorithm 1) is the (Euclidean) projection of \( x^{(n)} \) onto the set \( C \), and the inequality

\[
(x^{(n)} - z^{(n)})^\top x \leq (x^{(n)} - z^{(n)})^\top z^{(n)}
\]

or equivalently

\[
(x^{(n)} - z^{(n)})^\top (x - z^{(n)}) \leq 0
\]

(3.6)
added in the algorithm is a supporting valid inequality for $\mathcal{C}$ at the support point $z^{(n)}$. Let
\[
 f^{(n)}(x) := (x^{(n)} - z^{(n)})^\top (x - z^{(n)}),
\]
and \( \text{dist}(x^{(n)}, \mathcal{C}) := \min_{z \in \mathcal{C}} \|x^{(n)} - z\|_2 \). We have \( f^{(n)}(x^{(n)}) = (x^{(n)} - z^{(n)})^\top (x^{(n)} - z^{(n)}) = \text{dist}^2(x^{(n)}, \mathcal{C}) \). For any \( i, j \) satisfying \( j > i \geq i \), the following inequalities hold
\[
 f^{(n)}(x^{(n)_i}) = (x^{(n)_i} - z^{(n)_i})^\top (x^{(n)_j} - z^{(n)_i}) \\
= (x^{(n)_i} - z^{(n)_i})^\top (x^{(n)_i} - z^{(n)_i}) + (x^{(n)_i} - z^{(n)_i})^\top (x^{(n)_j} - x^{(n)_i}) \\
\geq (\text{dist}(x^{(n)_i}, \mathcal{C}))^2 - \|x^{(n)_i} - z^{(n)_i}\| \cdot \|x^{(n)_j} - x^{(n)_i}\| \\
= (\text{dist}(x^{(n)_i}, \mathcal{C}))^2 - \text{dist}(x^{(n)_i}, \mathcal{C}) \cdot \text{dist}(x^{(n)_j}, x^{(n)_i}) \\
\geq (\text{dist}(x^{(n)_i}, \mathcal{C}))^2 - \frac{1}{4} \text{dist}(x^{(n)_i}, \mathcal{C}) \cdot \text{dist}(\hat{x}, \mathcal{C}) \\
\geq \frac{9}{16} \text{dist}^2(\hat{x}, \mathcal{C}) - \frac{1}{4} \left[ \text{dist}(\hat{x}, \mathcal{C}) + \text{dist}(\hat{x}, x^{(n)_i}) \right] \cdot \text{dist}(\hat{x}, \mathcal{C}) \\
\geq \frac{9}{16} \text{dist}^2(\hat{x}, \mathcal{C}) - \frac{5}{16} \text{dist}^2(\hat{x}, \mathcal{C}) = \frac{1}{4} \text{dist}^2(\hat{x}, \mathcal{C}) > 0.
\]

Derivation in (3.8) uses the property \( \text{dist}(x^{(n)_i}, \mathcal{C}) \geq \frac{1}{2} \text{dist}(\hat{x}, \mathcal{C}) \) as \( x^{(n)_i} \to \hat{x} \), inequalities (3.5) and the triangle inequality \( \text{dist}(x, \mathcal{C}) \leq \text{dist}(x, y) + \text{dist}(y, \mathcal{C}) \) holding for any \( x, y \) and convex set \( \mathcal{C} \). The intuition behind (3.8) is to see whether valid inequality generated at an iteration \( n_i \) is satisfied by all the points in the subsequence generated afterward.

The (3.8) shows that \( x^{(n)_i} \) violates the separation cut \( f^{(n_i)}(x) \leq 0 \) generated at a previous iteration \( n_i \) \( (n_i < n_j) \), which leads to a contradiction since \( x^{(n)_i} \) is an optimal solution of (MS-\( n_j \)) which includes 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 finite number of iterations by solving a convex program.

**Lemma 3.2** Suppose Algorithm 2 does not terminate finitely, and let \( \{x^{(n)}\}_{i=1}^\infty \) be a convergent subsequence generated by this algorithm. Let \( x^\star \) be the limit point of this subsequence. Then, for a sufficiently large \( n_i \), \( x^\star \) is an optimal solution of the problem:
\[
\min_{x \in \mathcal{C}} c^\top x \\
\text{s.t. } Ax = b, \ x_j = x^{(n_i)}_j, \ j \in [l_1].
\]

Moreover, the condition at Line 27 of Algorithm 2 is satisfied for a sufficiently large \( n_i \).

**Proof** By Lemma 3.1, the limit point \( x^\star \) of the subsequence is in \( \text{bd}(\mathcal{C}) \). By Proposition 3.1 there exists an iteration index \( N \) such that \( x^{(n)}_j = x^\star_j \in \mathbb{Z} \)
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 (3.9) fixes the integer component to be $x_Z^{(n_i)}$. It follows that the convex program (3.9) 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

\[
\begin{align*}
U^{(n_i)} &= \min c^T x \quad \text{s.t.} \quad Ax = b, \ x_j = x_j^{(n_i)}, \ j \in [l_1], \ x \in \mathbb{C}, \quad (3.10) \\
U^* &= \min c^T x \quad \text{s.t.} \quad Ax = b, \ x_j = x_j^*, \ j \in [l_1], \ x \in \mathbb{C}. \quad (3.11)
\end{align*}
\]

We now show that $x^*$ is an optimal solution of (3.10) and (3.11). If not, then assume that $\hat{x}$ is an optimal solution of (3.11), and it satisfies $c_R \hat{x}_R < c_R x_R^*$. Obviously, $(x_Z^{(n_i)}, \hat{x}_R) = (x_Z^*, \hat{x}_R)$ is a feasible solution of (3.11) for sufficiently large $n_i$. Moreover, we have

\[
c^T (x_Z^{(n_i)}, \hat{x}_R) = c^T x_Z^{(n_i)} + c^T x_R^* - c^T (x_R^* - \hat{x}_R) < c^T x_Z^{(n_i)} + c^T x_R^*.
\]

Since $\lim_{i \to \infty} x_Z^{(n_i)} = x_Z^*$, the above inequality implies that $c^T (x_Z^{(n_i)}, \hat{x}_R) < c^T x_Z^{(n_i)}$ 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 (3.11) 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 (3.10). The above argument also indicates that the condition at Line 20 of Algorithm 1 is satisfied for any sufficiently large iteration since $x^* \in \text{bd}(\mathcal{C})$ is a solution satisfying this condition.

**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 $\{x^{(n_i)}\}_{n_i=0}^{\infty}$ 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 $x^* \in \text{bd}(\mathcal{C})$. We focus on proving that $L^{(n_i)} = U^{(n_i)}$ for a sufficiently large $n_i$, and hence the algorithm should terminate (Line 17).

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

\[
U^* = \min c^T x \\
\quad \text{s.t.} \quad Ax = b, \ x_Z = x_Z^*, \ x \in \mathcal{C}. \quad (3.12)
\]
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:

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

(3.13)

Note that the constraints \( C^{(k)} x \leq d^{(k)} \) for all \( k \in I^{(n_i - 1)} \) are the supporting valid inequalities generated at Step 3 when \( \bar{x}^{(k)} \in \text{bd}(C) \) and \( -c \in N_{S^{(k)}}(\bar{x}^{(k)}) \) (Algorithm 1, Line 22). These supporting valid inequalities pass through the point \( \bar{x}^{(k)} \in \text{bd}(C) \), which is an optimal solution of (3.12). Therefore, the following relation holds:

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

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

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

\[
= \min c^\top x \quad \text{s.t.} \quad \begin{cases} 
Ax = b, & x_Z = x_Z^*, \quad x \in \mathbb{R}^{l_1 + l_2}, \quad x \in C, \\
C^{(k)} x \leq d^{(k)} & \forall k \in I^{(n_i - 1)} \end{cases}
\]

\[
= U^* = U^{(n_i)},
\]

where we use Proposition 2.2 to replace \( C^{(k)} x \leq d^{(k)} \) with \( x \in C \) in the third equality in the above analysis. This concludes the proof.

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

We now focus on developing a finitely convergent cutting-plane decomposition algorithm for (JMIPC). The general idea of this algorithm is as follows: The problem (JMIPC) is decomposed into a first-stage master problem in the space of \([x, \eta]\) and a second-stage in the space of \(y\), where \(\eta\) is an auxiliary variable that linking the first-stage and the second-stage problems. In particular, the
master problem at the beginning of iteration $m$ is formulated as the following mixed 0-1 linear program:

$$\min_{x \in X \cap \{0, 1\}^l} \quad c^T x + \eta$$

$$\text{subject to} \quad \eta \geq a^k x + b^k \quad \forall k \in [m - 1],$$

where $\eta \geq a^k x + b^k$ is a valid inequality added to the first-stage at a previous iteration $k$. The details on generation of these inequalities are given later. The algorithm calls the cutting-plane oracle from [8] to solve the master problem and obtain the first-stage solution $x^m$.

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

$$\min_{y \in Y \cap (\mathbb{Z}^l_2 \times \mathbb{R}^l_3)} \quad h^T y$$

$$\text{subject to} \quad Tx^m + Wy = q,$$

$$g_i(x^m, y) \leq 0 \quad \forall i \in I.$$  \hspace{1cm} (4.2)

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 (4.2), such that upon termination the parametric cutting-plane algorithm generates the following linear relaxation problem of (4.2):

$$\min_{y \in \mathbb{R}^l_1 \times \mathbb{R}^l_2} \quad h^T y$$

$$\text{subject to} \quad Tx^m + Wy = q,$$

$$C^m x^m + D^m y \leq F^m,$$  \hspace{1cm} (4.3)

where the matrices satisfy the following conditions:

- There exists a common optimal solution (denoted as $y^m$) of (4.2) and (4.3);
- The linear inequalities $C^m x^m + D^m y \leq F^m$ are valid for the original feasible set $\{[x, y] : Tx + Wy = q, g_i(x, y) \leq 0 \forall i \in I, x \in X \cap \{0, 1\}^l, y \in Y \cap (\mathbb{Z}^l_2 \times \mathbb{R}^l_3)\}$ of (JMCP).

Let $\mu^m$ and $\lambda^m$ be the optimal dual vectors corresponding to the equality and inequality constraints in (4.3) ($\mu^m$ and $\lambda^m$ satisfies the constraints $W\mu^m = h, \quad D^m \lambda^m + h = 0$ and $\lambda^m \geq 0$). We can then generate the following Benders’ cut using strong duality:

$$\eta \geq (\lambda^m C^m - \mu^m T) x - (\lambda^m F^m - \mu^m q).$$  \hspace{1cm} (4.4)

The above inequality will serve as an additional inequality added to the master problem (4.1) at the end of iteration $m$, i.e., $\eta \geq a^m x + b^m$ with $a^m = (\lambda^m C^m - \mu^m T)$ and $b^m = -(\lambda^m F^m - \mu^m q)$.

The core of the decomposition algorithm is to generate the linear relaxation (4.3) of (4.2) 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 results of convergence 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

$$\min_{y \in \mathcal{Y} \cap \mathbb{Z}_{l_2} \times \mathbb{R}_{l_3}} h^\top y$$

$$\text{s.t. } Tx + Wy = q,$$

$$g_i(\hat{x}, y) \leq 0 \quad \forall i \in \mathcal{I},$$

(PMICP)

where $\hat{x} \in \mathcal{X} \cap \{0, 1\}^{l_1}$ is a first stage solution passed to the problem as a parameter like in (4.2). The $\hat{x}$ is fixed when solving (PMICP) but it is subject to change in following-up second-stage problems.

The high-level idea of this algorithm is to first linearize the convex constraints in (PMICP) to get the following parametric mixed-integer linear program which is a relaxation of (PMICP):

$$\min_{y \in \mathbb{Z}_{l_2} \times \mathbb{R}_{l_3}} h^\top y$$

$$\text{s.t. } Tx + Wy = q,$$

$$T'\hat{x} + W'y \leq q',$$

(4.5)

where $\hat{x} \in \mathcal{X} \cap \{0, 1\}^{l_1}$ is the problem parameter, and $(T', W', q')$ are some matrices induced by linearization. It then solves (4.5) using the parametric cutting-plane oracle developed in [21]. These two steps form a main iteration in the algorithm.

More specifically, 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

$$\min_{y \in \mathbb{Z}_{l_2} \times \mathbb{R}_{l_3}} h^\top y$$

$$\text{s.t. } Tx + Wy = q,$$

$$a^{(k)}(\hat{x}) + b^{(k)}y + c^{(k)} \leq 0 \quad \forall k \in [n - 1]$$

$$C^{(k)}\hat{x} + D^{(k)}y \leq d^{(k)} \quad \forall k \in \mathcal{I}^{(n-1)},$$

(PMS-n)

where $a^{(k)}(\hat{x}) + b^{(k)}y + c^{(k)} \leq 0$ is the cutting plane generated at iteration $k$ induced by a projection problem (referred as projection cuts), and $C^{(k)}\hat{x} + D^{(k)}y \leq d^{(k)}$ are outer-approximation cuts generated similarly as described in Section 2. Note that (PMS-n) is in the form (4.5). We now highlight the differences of generating the projection cuts and the outer-approximation cuts for the parametric problem.

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

$$\min ||x - \hat{x}||^2 + ||y - y^{(n)}||^2$$

$$\text{s.t. } g_i(x, y) \leq 0 \quad \forall i \in \mathcal{I}, \quad y \in \mathcal{Y}.$$  

(4.6)
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 normal vector \([\hat{x} - x^*, y^{(n)} - y^*]\). Specifically, we have

\[
\begin{align*}
a^{(n)} &= \hat{x} - x^*, \\
b^{(n)} &= y^{(n)} - y^*, \\
c^{(n)} &= -(\hat{x} - x^*)^\top x^* - (y^{(n)} - y^*)^\top y^*.
\end{align*}
\] (4.7)

Note that the general form of the projection cut should be

\[
\begin{align*}
a^{(n)}^\top x + b^{(n)}^\top y + c^{(n)} &\leq 0. \\
\end{align*}
\] (4.8)

We set \(x = \hat{x}\) and add the above inequality to update (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:

\[
\begin{align*}
\min & \quad h^\top y \\
\text{s.t.} & \quad T\hat{x} + Wy = q, \\
& \quad g_i(\hat{x}, y) \leq 0 \quad \forall i \in I, \\
& \quad y = y^{(n)}, \quad y \in Y,
\end{align*}
\] (4.9)

where \(y_Z\) represent the integer variables in \(y\). Recall that non-linear convex set of (JMICP) is denoted as:

\[
S := \{ [x, y] : g_i(x, y) \leq 0 \forall i \in I, \ y \in Y \}.
\] (4.10)

Let problem (4.9) has non-empty relative interior and \(y^*\) be an optimal solution of it. Then, similar to cases following (2.8), either \([\hat{x}, y^*]\) is in the interior of \(S\) since by Assumption 2 its interior cannot be empty or the point \([\hat{x}, y^*]\) is on the boundary of the set \(S\) due to linearity of the objective of (4.9).

**Lemma 4.1** Let \(y^*\) be an optimal solution of (4.9). If the point \([\hat{x}, y^*]\) \(\in bd(S)\), then

(1) the following relation holds:

\[
-h \in N_W(y^*) + \sum_{i \in I_0} \text{cone}(\partial_y g_i(\hat{x}, y^*)) + \text{span}([1_Z, 0_R]) + N_Y(y^*),
\] (4.11)

where \(N_W(y^*)\) is the normal cone of \(\{y : Wy = q - T\hat{x}\}\) at \(y^*\), \(I_0 \subseteq I\) is the index set of active constraints in \(\{g_i : i \in I\}\) at the point \([\hat{x}, y^*]\), \([1_Z, 0_R]\) is the vector that has all ones at the integer components and all zeros at other components of the vector \(y\), and \(N_Y(y^*)\) is the normal cone of \(Y\) at \(y^*\). The addition follows the Minkowski rule.

(2) One can find vectors \(\gamma \in N_W(y^*), \ r_i \in \text{cone}(\partial_y g_i(\hat{x}, y^*)) \forall i \in I_0, \ a\) number \(\alpha\), and a vector \(r' \in N_Y(y^*)\) such that

\[
-h = \gamma + \sum_{i \in I_0} r_i + \alpha[1_Z, 0_R] + r'.
\] (4.12)
The optimal objective of \((4.9)\) is equal to the optimal objective of the following linear program:

\[
\begin{align*}
\min \quad & h^\top y \\
\text{s.t.} \quad & W y = q - T\hat{x}, \\
& r_i^\top (y - y^*) \leq 0 \quad \forall i \in I_0, \\
& yZ = y_Z^{(n)}, \\
& r^\top (y - y^*) \leq 0.
\end{align*}
\]

Proof (1) Note that \((4.11)\) is exactly the optimality condition of the problem \((4.9)\) at \(y = y^*\), which should obviously hold. (2) The oracle provided in Proposition 2.1 can generate the desired \(\gamma, r_i (i \in I_0), \alpha\) and \(r^\prime\) by realizing that \(\text{span}([1_Z, 0_R]) = \text{cone}([1_Z, 0_R]) + \text{cone}([-1_Z, 0_R])\). (3) We first observe that \(y = y^*\) is a feasible solution to \((4.13)\). Furthermore, the optimality condition of the linear program is exactly the same as \((4.12)\), 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 \(r_i \in \text{cone}(\partial y g_i(\hat{x}, y^*))\) and \(i \in I_0\) of the linear program on the right side of \((4.13)\) can be alternatively written as \(r_i^\top (x - \hat{x}) + r_i^\top (y - y^*) \leq 0\) for some \(r_x \in \partial x g_i(\hat{x}, y)\), and then set \(x = \hat{x}\). Notice that the later inequalities in the \([x, y]\)-space are valid for the set \(\mathcal{S}\), which is guaranteed by Assumption 3. The later representation is used to generate a Benders’ cut for the first-stage master problem.

Lemma 4.1 indicates that one can obtain \(r_i \in \text{cone}(\partial y g_i(\hat{x}, y^*))\) for every \(i \in I_0\) where \(I_0\) is the index set of active constraints at \([\hat{x}, y^*]\), and \(r^\prime \in \mathcal{N}_Y(y^*)\), such that the optimal value of \((4.9)\) is equal to the optimal value of the following linear program when setting \(x = \hat{x}\):

\[
\begin{align*}
\min \quad & h^\top y \\
\text{s.t.} \quad & Tx + W y = q, \\
& r^\top x,i(x - \hat{x}) + r_i^\top (y - y^*) \leq 0 \quad \forall i \in I_0, \\
& r^\top (y - y^*) \leq 0, \\
& yZ = y_Z^{(n)},
\end{align*}
\]

where \(r^\top x,i\) is an arbitrary element chosen from \(\partial x g_i(\hat{x}, y^*)\) for every \(i \in I_0\) (Remark 4.1). Let the following generic form

\[
C^{(n)} x + D^{(n)} y \leq d^{(n)}
\]

represent the supporting inequalities

\[
\begin{align*}
& r^\top x,i(x - \hat{x}) + r_i^\top (y - y^*) \leq 0 \quad \forall i \in I_0, \\
& r^\top (y - y^*) \leq 0,
\end{align*}
\]

that are involved in \((4.14)\).
In Algorithm 2, the inequalities \( (4.15) \) are added into \( (\text{PMS-}n) \) at the end of iteration \( n \). It is shown in Theorem 4.1 that \( (4.15) \) are valid for all \( [x, y] \) in \( S \). The pseudo code of the generalized parametric cutting-plane algorithm for solving \( (\text{PMS-}n) \) is given in Algorithm 2.

**Theorem 4.1** Suppose Assumptions 3 and 4 hold. 
(1) 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 \( (\text{PMS-}n) \) are valid for points in the set \( S \) defined in \( (4.10) \). 
(2) The generalized parametric cutting-plane algorithm (Algorithm 2) can solve the parametric mixed-integer convex program \( (\text{PMICP}) \) to optimality in a finite number of iterations.

**Proof** (1) First, we notice that all cuts added to the master problem \( (\text{PMS-}n) \) are used to build an outer-approximation of the set \( S \), which can be seen from how these cuts are generated. Therefore, they are valid for all points in \( S \). 
(2) It remains to show that the algorithm converges to an optimal solution of \( (\text{PMICP}) \) in a finite number of iterations. If not, there exists a convergent subsequence denoted as \( \{[\hat{x}, y^{(n_k)}]\}_{k=1}^\infty \) which converges to a point \( [x^*, y^*] \). Similarly as in the proof of Theorem 3.1, one can show that \( [\hat{x}, y^*] \) is an optimal solution of \( (\text{PMS-}n) \), i.e., \( y_2^{(n_k)} = y_2^* \) for any sufficiently large \( n_k \), and also \( [\hat{x}, y^*] \) must be a point in the sub-sequence \( \{[\hat{x}, y^{(n_k)}]\}_{k=1}^\infty \) due to \( (4.14) \). This indicates that the algorithm must terminate in a finite number of iterations.

We now discuss the need of Assumption 3 in generating a valid parametric cut. We deliver some insights about the assumption using a specific example. Suppose \( [\hat{x}, y^*] \) is on the boundary of the set \( S \). For simplicity, let us assume that \( g_i^*(x, y) \leq 0 \) is the only active constraint at \( [\hat{x}, y^*] \), i.e., \( g_i^*(\hat{x}, y^*) = 0 \), \( g_i(\hat{x}, y^*) < 0 \) for all \( i \in I \setminus \{i^*\} \) and \( y^* \in \text{int}(Y) \). The necessary and sufficient condition for an inequality

\[
a^\top(x - \hat{x}) + b^\top(y - y^*) \leq 0
\]

(4.18)
to be valid for all points in the set \( S \) is \((a, b) \in N_S(\hat{x}, y^*) \) (the normal cone of \( S \) at \( [\hat{x}, y^*] \)), which is equivalent to \((a, b) \in \partial g_i^*(\hat{x}, y^*) \) in this case. When we fix \( \hat{x} \) and solve \( (4.9) \) to get \( y^* \), the optimality condition is written as

\[-h \in N_W(y^*) + \text{cone}(\partial g_i^*(\hat{x}, y^*)) + \text{span}([1, 0]) \]

To get an equivalent linear program of \( (4.9) \), we generate a vector \( b \in \partial g_i^*(\hat{x}, y^*) \) and a number \( \alpha \) using the oracle from Proposition 2.1 such that \( h = \gamma + b + \alpha[1, 0] \). To lift a valid inequality generated in the space of \( y \) (fixing \( x = \hat{x} \)) to the space of \( [x, y] \) and maintaining its validness, requires existence of an \( a \) such that \((a, b) \in \partial_{[x, y]} g_i^*(\hat{x}, y^*) \). 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 3 makes the lift of an inequality from the \( y \)-space to \([x, y]\)-space achievable.
Algorithm 2 An algorithm for solving a parametric mixed-integer convex program (PMICP).

1: Set \( n \leftarrow 1 \), \( \text{flag} \leftarrow 0 \) and \( I(0) \leftarrow \emptyset \).
2: while \( \text{flag} = 0 \) do
3: (Start iteration \( n \).)
4: \textbf{Step 1:} Solve the master problem (PMS-\( n \)) using the cutting-plane oracle from [11].
5: to get an optimal solution \( y^{(n)} \) and optimal objective denoted as \( L^{(n)} \).
6: Let \( y^{(n)}_Z \in \mathbb{Z}^l \) and \( y^{(n)}_R \in \mathbb{R}^l \).
7: if \( \hat{x}, y^{(n)} \in S \) then
8: Stop and return \( y^{(n)} \) as an optimal solution of (PMICP);
9: end if
10: \textbf{Step 2:} Solve (4.6) to get the projection point \( [\hat{x}, y^{(n)}] \) on \( S \) (defined in (4.10)).
11: Add the separation (projection) cut \( a^{(n)} \top \hat{x} + b^{(n)} \top y + c^{(n)} \leq 0 \) (defined in (4.8)) to (PMS-\( n \)).
12: \textbf{Step 3:} Solve the continuous convex program (4.9).
13: if (4.9) is feasible then
14: Let \( U^{(n)} \) be the optimal value of (4.9).
15: if \( L^{(n)} = U^{(n)} \) then
16: Stop and return an optimal solution of (4.9) as an optimal solution of PMS-\( n \).
17: end if
18: \textbf{if} (4.9) has a solution \( y^* \) such that \( [\hat{x}, y^*] \in \text{bd}(S) \) \textbf{then}
19: Generate a finite number of supporting valid inequalities as defined in (4.15): \( C^{(n)} \hat{x} + D^{(n)} y \leq d^{(n)} \) (4.19)
20: and add them to (PMS-\( n \)).
21: \textbf{else}
22: Set \( I^{(n)} \leftarrow I^{(n-1)} \cup \{n\} \).
23: \textbf{end if}
24: \textbf{else}
25: \textbf{else}
26: Set \( I^{(n)} \leftarrow I^{(n-1)} \).
27: \textbf{end if}
28: \( \text{flag} \leftarrow 1 \).
29: \textbf{end while}
30: Return \( y^{(n)} \).

4.2 A decomposition cutting-plane algorithm for (JMIPC)

We now focus on developing a cutting-plane decomposition algorithm for solving (JMIPC) to optimality. The master problem (4.1) can be iteratively updated by adding Benders’ cuts generated from solving the second-stage problem (4.2). Specifically, Algorithm 2 is applied to solve (4.2) to optimality, and when the algorithm terminates, it generates a mixed-integer linear program (MILP) that has a same optimal solution as (4.2). The MILP is solved to optimality with the cutting-plane oracle from [11], and the oracle generates a linear program whose optimal solution is the same as that of the MILP. Suppose the linear program at the main iteration \( m \) is denoted as (4.3). Then the Benders’ cut (4.4) 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, 3 and 4 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:

\[
h^\top y \geq a^m x + b^m \quad \forall m,
\]

where the coefficients \( a^m \) and \( b^m \) are from the Bender’s inequality \( \eta \geq a^m x + b^m \) in the master problem (4.1) for solving (JMICP). The Bender’s inequality can also be represented as (4.4). Recall that when the generalized parametric cutting-plane algorithm (Algorithm 2) is applied to solve the second-stage problem (4.2), it repeatedly solves a master problem (PMS-\( n \)), and it adds convexification cuts to update master problem at every iteration. By Theorem 4.1, the convexification cuts added to the master problem are valid for the set \( \mathcal{S} \). It implies that the linear program obtained upon termination of Algorithm 2 has the form

\[
\min h^\top y \quad \text{s.t.} \quad Tx + Wy = q, C^m x^m + D^m y \leq F^m,
\]

where \( C^m, D^m, F^m \) are some coefficient matrices as in (4.3). Then it follows that the Benders’ cut \( \eta \geq a^m x + b^m \) generated from the linear program should be valid for the set \( \{ [x, y, \eta] : h^\top y \leq \eta, Tx + Wy = q, g_i(x, y) \leq 0 \forall i \in I, y \in Y \} \).

For the first-stage solution \( x^m \) obtained at the main iteration \( m \) of Algorithm 3 let \( \text{obj}_2^m(x^m) \) denote the optimal objective of the second-stage problem (4.2) at \( x = x^m \). Strong duality implies that \( \text{obj}_2^m(x^m) = a^m x^m + b^m \). 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 in a finite number of iterations. Since the feasible set \( \mathcal{X} \cup \{0, 1\}^\ell \) is finite, there exist \( m \) and \( n \) (\( m > n \)) such that \( x^m = x^n \).

We have

\[
U^m = c^\top x^m + \text{obj}_2^m(x^m) \\
\geq L^m = c^\top x^m + \eta^m \\
\geq c^\top x^m + a^n x^m + b^n \\
= c^\top x^n + a^n x^n + b^n \\
= c^\top x^n + \text{obj}_2^n(x^n) \\
= U^n,
\]

where we use the fact that \( \eta^m \geq a^n x^n + b^n \) since \( \eta \geq a^n x + b^n \) is a constraint in the master problem (4.1) at the main iteration \( m \) of Algorithm 3. The above inequalities indicate that \( U^m = L^m = U^n \) and hence the algorithm should terminate at or before the main iteration \( m \).
Algorithm 3 A decomposition cutting-plane algorithm for (JMCP).

1: Set $m \leftarrow 1$, $L \leftarrow -\infty$ and $U \leftarrow \infty$.
2: while $L < U$ do
3: Solve the current first-stage problem (4.1) for the main iteration $m$ using the cutting-plane oracle from [11], and get the optimal solution $x^m$.
4: Set $L \leftarrow \operatorname{obj}(4.1)$, where $\operatorname{obj}(4.1)$ represents the optimal objective of (4.1).
5: Substitute $x^m$ into the parametric MICP (4.2), and apply Algorithm 2 to solve it.
6: Set $U \leftarrow c^\top x^m + \operatorname{obj}(4.2)$.
7: When Algorithm 2 terminates with an optimal solution to (4.2), a linear program (4.3) is established. Generate a Benders’ cut (4.4) from the dual objective of (4.3), and add it to (4.1) as $\eta \geq a^m \top x + b^m$.
8: Set $m \leftarrow m + 1$.
9: end while
10: return $x^m$ as an optimal solution of (JMCP).

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

The decomposition cutting-plane algorithm (Algorithm 3) can be extended to solve the following class of distributionally-robust two-stage mixed-integer convex programs with finite convergence:

$$
\begin{align*}
\min_x & \quad c^\top x + \max_{P \in \mathcal{P}} \mathbb{E}_{\xi \sim P}[Q(x, \xi)] \\
\text{subject to} & \quad Ax \leq b, \ x \in \mathcal{C}, \ x \in \{0, 1\}^{l_1}
\end{align*}
$$

(TSS-MICP)

where $\mathcal{C}$ is a convex set for the first-stage variable, $Ax \leq b$ defines a polytope (i.e., bounded) $\xi$ is a random vector having a finite support denoted as $\{\xi^\omega | \omega \in \Omega\}$ ($\Omega$ is a finite set of scenarios), $\mathcal{P}$ is a candidate probability distribution of $\xi$, $\mathcal{P}$ is the ambiguity set of all candidate probability distributions, and $Q(x, \omega)$ is the recourse function at the scenario $\omega \in \Omega$, where $\Omega$ is a finite set. We further assume that the ambiguity set $\mathcal{P}$ is a convex set in $\mathbb{R}^{l_2}$. The recourse function $Q(x, \xi^\omega)$ is given as

$$
\begin{align*}
Q(x, \xi^\omega) = \min_{y^\omega} & \quad h^\omega \top y^\omega \\
\text{subject to} & \quad T^\omega x + W^\omega y^\omega = q^\omega \\
& \quad g_i^\omega(x, y^\omega) \leq 0 \quad \forall i \in I, \\
& \quad y^\omega \in \mathcal{Y} \cap (\mathbb{Z}^{l_2} \times \mathbb{R}^{l_3}),
\end{align*}
$$

(5.1)

where every $g_i^\omega$ ($i \in I$, $\omega \in \Omega$) is a smooth convex function in $[x, y^\omega]$. We denote the minimization problem in (5.1) as $\text{Sub}(x, \omega)$. 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 $\omega$. Consider applying Algorithm 2 to solve this sub-problem (a parametric MICP). When the algorithm terminates, cuts have been added. Suppose the linear program at the termination is given
\[
\min_{y^\omega \in \mathbb{R}^{l_2+l_3}} h^\omega y^\omega
\]
\[
\text{s.t. } W^\omega y^\omega = q^\omega - T^\omega \hat{x}, \quad \text{dual } \theta^\omega
\]
\[
Q^\omega y^\omega \geq s^\omega - R^\omega \hat{x}, \quad \text{dual } \mu^\omega \geq 0
\] (5.2)

where \(Q^\omega, R^\omega\) and \(s^\omega\) are appropriate matrices and vectors of coefficients that represent the complete set of cutting planes at termination. The LP relaxation should give the same objective as \((5.1)\). This property is formally stated as follows:

Lemma 5.1 When Algorithm 2 is applied to solve the parametric MICP problem \(\text{Sub}(x, \omega)\), the termination criteria and Theorem 3.1 ensure that the optimal objective of \((5.2)\) must be equal to \(Q(\hat{x}, \xi^\omega)\).

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 \(\mathbb{R}^{n_\Omega}\))

\[
\max_{p \in \mathcal{P}} \sum_{\omega \in \Omega} p^\omega Q(\hat{x}, \xi^\omega)
\] (5.3)

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 \(W^\omega y^\omega = q^\omega - T^\omega \hat{x}, Q^\omega y^\omega \geq s^\omega - R^\omega \hat{x}\) and the worst-case probability distribution \(p\). Specifically, we can construct the following aggregated Benders’ cut:

\[
\eta \geq -\left(\sum_{\omega \in \Omega} p^\omega \mu^\omega R^\omega + \sum_{\omega \in \Omega} p^\omega \theta^\omega T^\omega\right) x + \left(\sum_{\omega \in \Omega} p^\omega \mu^\omega s^\omega + \sum_{\omega \in \Omega} p^\omega \theta^\omega q^\omega\right),
\] (5.4)

where \(p^\omega\) is the probability of scenario \(\omega\) obtained from solving \((5.3)\). This cut is added to the first-stage master problem of \((\text{TSS-MICP})\). Let \(r^\omega = -\left(\mu^\omega R^\omega + \theta^\omega T^\omega\right)^T, u^\omega = (\mu^\omega s^\omega + \theta^\omega q^\omega)\). So at the \(m^{th}\) main iteration, the first-stage master problem of \((\text{TSS-MICP})\) is given as

\[
\min_{x \in \mathcal{C} \cap \{0, 1\}^l} c^T x + \eta
\]
\[
\text{s.t. } Ax \leq b,
\]
\[
\eta \geq \sum_{\omega \in \Omega} p^{k\omega} k^{\omega T} x + \sum_{\omega \in \Omega} p^{k\omega} u^{k\omega} \quad \forall k \in [m-1],
\] (5.5)

where \(\eta \geq \sum_{\omega \in \Omega} p^{k\omega} k^{\omega T} x + \sum_{\omega \in \Omega} p^{k\omega} u^{k\omega}\) 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 \((5.5)\) is a mixed 0-1 linear convex program which can again be solved using Algorithm 4. The first-stage solution of \((5.5)\) (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
Algorithm 4 A decomposition cutting-plane algorithm for solving (TSS-MICP).

1: Initialization: $L \leftarrow -\infty$, $U \leftarrow \infty$, $m \leftarrow 1$.
2: while $U - L > 0$ do
3:   Solve the first-stage problem (5.5) to optimality with main iteration index being $m$.
4:   Let $(\eta^m, x^m)$ be the optimal solution.
5:   Update the lower bound as $L \leftarrow c^\top x^m + \eta(m)$.
6:   Set the current best solution as $x^* \leftarrow x^m$.
7:   for $\omega \in \Omega$ do
8:      Solve the second-stage problem Sub$(x^m, \omega)$ to optimality using Algorithm 2.
9:      When Algorithm 2 terminates, it generates a LP relaxation (5.2) with $\hat{x} = x^m$.
10:     Get the optimal values of the dual variables of (5.2) and denote them as $\lambda^m\omega$ and $\mu^m\omega$.
11:    end for
12:   Obtain the worst-case probability distribution $p^m = \{p^m\omega \mid \omega \in \Omega\}$
13:   by solving (5.3) with $\hat{x} = x^m$.
14:   Generate the following aggregated Benders’ cut with iteration index $m$:
15:      $\eta \geq \sum_{\omega \in \Omega} p^m\omega \lambda^m\omega x + \sum_{\omega \in \Omega} p^m\omega \mu^m\omega$.
16:   Add the Benders’ cut to the first-stage problem (5.5).
17:   Update the upper bound as $U \leftarrow \min\{U, c^\top x^m + \sum_{\omega \in \Omega} p^m\omega Q(x^m, \xi^\omega)\}$.
18: $m \leftarrow m + 1$.
19: end while
20: Return $x^*$.

Theorem 5.1 Suppose (TSS-MICP) has a complete recourse. Let Assumption [3] hold for every $g^\omega$ and Assumption [2] hold. Algorithm 4 returns an optimal solution of (TSS-MICP) after finitely many iterations.

Proof The proof is similar to the proof of Theorem 2.1 in [21]. Let the first-stage problem at main iteration $k$ be denoted by Master-$k$. Lemma 5.1 and the strong duality of (5.2) imply that $Q(x^k, \omega) = x^k\lambda^\top x^k + u^k\omega$. Therefore, we have

$$G(x^k) := \max_{p \in P} E_P[Q(x^k, \xi)] = \sum_{\omega \in \Omega} p^k\omega x^k + \sum_{\omega \in \Omega} p^k\omega u^k\omega.$$  (5.6)

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

$$U^{k_1} = c^\top x^{k_1} + \sum_{\omega \in \Omega} p^{k_1\omega} Q(x^{k_1}, \xi^\omega) = c^\top x^{k_1} + G(x^{k_1}),$$  (5.7)
where (5.6) is used to obtain the last equation. The optimal value of Master-$k_2$ gives a lower bound
\[ L^{k_2} = c^T x^{k_2} + \eta^{k_2}. \]
Since $k_2 > k_1$, Master-$k_2$ has the following constraint:
\[ \eta \geq \sum_{\omega \in \Omega} p^{k_1 \omega} x^{k_1 \omega} + \sum_{\omega \in \Omega} p^{k_1 \omega} u^{k_1 \omega}. \]  
(5.8)

Therefore, we conclude that
\[ L^{k_2} = c^T x^{k_2} + \eta^{k_2} \]
\[ \geq c^T x^{k_2} + \sum_{\omega \in \Omega} p^{k_1 \omega} x^{k_1 \omega} + \sum_{\omega \in \Omega} p^{k_1 \omega} u^{k_1 \omega} \]
\[ = c^T x^{k_1} + \sum_{\omega \in \Omega} p^{k_1 \omega} x^{k_1 \omega} + \sum_{\omega \in \Omega} p^{k_1 \omega} u^{k_1 \omega} \]
\[ = c^T x^{k_1} + G(x^{k_1}) = U^{k_2} \geq U^{k_2}, \]
(5.9)

where we use the fact that $x^{k_1} = x^{k_2}$, and inequalities (5.7)–(5.8) to obtain (5.9). 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 finite convergent linearize-branch-and-union decomposition algorithm

Note that Algorithm 2 requires repeatedly solving the master problem (PMS-$n$) using a pure cutting-plane oracle. This step is impractical since the number of cuts can be exponentially depending on the problem-size parameters. Furthermore, adding too many cuts may slow down the process of solving LP relaxations. To make the algorithm more suitable for implementation, we can make some modification by leveraging the branch-and-bound technique, while still achieving finite convergence for eventually solving (TSS-MICP).

Modified-Algorithm 2:

Step 1: Solve the master problem (PMS-$n$) using a branch-and-bound algorithm to get an optimal solution $y^{(n)}$ and optimal objective denoted as $L^{(n)}$.

Step 2 and Step 3 are exactly the same as Algorithm 2.

Proposition 6.1 Suppose Assumptions 3 and 4 hold. The Modified-Algorithm 2 can solve (PMICP) to optimality in a finite number of iterations.

Proof Although Modified-Algorithm 2 replaces the cut generation step with a branch-and-bound step to solve parametric MILP problem (master problem), convexification cuts generated in Step 2 and Step 3 only depend on the optimal solution $y^{(n)}$ of the master problem, which is independent of how the master problem is solved. Therefore, the proof of finite convergence is exactly the same as the proof of Theorem 4.1.
Lemma 6.1 Consider solving \((\text{PMICP})\) using the Modified-Algorithm 2. The linear program induced by any node \(\nu\) in the branch-and-bound tree generated for solving \((\text{PMS-n})\) can be written in the following form:

\[
\begin{align*}
\min_y & \quad h^\top y \\
\text{s.t.} & \quad T\hat{x} + Wy - q = 0, \\
& \quad R_\nu\hat{x} + Q_\nu y + s_\nu \leq 0, \\
& \quad y \in \mathbb{R}^{l_2+l_3},
\end{align*}
\]

(6.1)

where \(R_\nu\), \(Q_\nu\) and \(s_\nu\) are some coefficient matrices depending on the node \(\nu\).

Proof First we note that equality constraints are independent of any node \(\nu\) of the branch-and-bound tree. Next we note that in the Step 1 of Modified-Algorithm 2 the branch-and-bound procedure only adds branching inequalities such as \(y_i \leq b\) or \(y_i \geq b\) for some index \(i \in [l_2]\) and integer \(b\) to a new node in the branch-and-bound tree. These branching inequalities are obviously in the desired form of second set of constraint in (6.1). All the other inequalities created in Step 2 and Step 3 of Modified-Algorithm 2 are like \(a^{(k)}\hat{x} + b^{(k)}y + c^{(k)} \leq 0\) and \(C^{(n)}\hat{x} + D^{(n)}y \leq d^{(n)}\), which can also be represented by the form of constraints in (6.1).

Consider solving the problem \((\text{JMICP})\) in a decomposed manner similar to the approach developed in Section 4 but using a different method (branch-and-bound method) for solving the parametric mixed-integer linear program. For a given input first-stage solution \(x = \hat{x}\), when \((\text{PMICP})\) has been solved to optimality using the Modified-Algorithm 2 we consider the set \(\mathcal{L}(\hat{x})\) of leaf nodes of the branch-and-bound tree generated for solving the master problem \((\text{PMS-n})\) in the last main iteration of the Modified-Algorithm 2. Applying Lemma 6.1 we can get the following LP formulation of each node \(\nu \in \mathcal{L}(\hat{x})\):

\[
\begin{align*}
\min_y & \quad h^\top y \\
\text{s.t.} & \quad T\hat{x} + Wy - q = 0, \\
& \quad R_\nu\hat{x} + Q_\nu y + s_\nu \leq 0, \\
& \quad y \in \mathbb{R}^{l_2+l_3}.
\end{align*}
\]

(6.2)

Let \(y_\nu\) be the optimal solution of (6.2). Define the polytope \(P_\nu\) as follows:

\[
P_\nu := \left\{ x \mid T\hat{x} + Wy_\nu - q = 0, R_\nu\hat{x} + Q_\nu y_\nu + s_\nu \leq 0, \quad 0 \leq x \leq 1 \right\},
\]

(6.3)

where \(1\) represents a \(l_1\)-dimensional vector with every entry being 1. Let \(\Pi(\hat{x}) = \bigcup_{\nu \in \mathcal{L}(\hat{x})} P_\nu\) be the union of polytopes (6.3) induced by all nodes in \(\mathcal{L}(\hat{x})\).
Using the lift-and-project technique \cite{5}, the convex hull \( \text{conv}(\Pi(\hat{x})) \) can be represented as

\[
\text{conv}(\Pi(\hat{x})) = \text{Proj}_x \left\{ x, x_\nu, \alpha_\nu \mid \forall \nu \in \mathcal{L}(\hat{x}) \right\}
\]

\[
\begin{align*}
&\sum_{\nu \in \mathcal{L}(\hat{x})} \alpha_\nu = 1, \\
x &= \sum_{\nu \in \mathcal{L}(\hat{x})} x_\nu, \\
0 &\leq x_\nu \leq 1, \\
\alpha_\nu &\geq 0 \quad \forall \nu \in \mathcal{L}(\hat{x}).
\end{align*}
\]  

(6.4)

We can parameterize all the valid inequalities for \( \text{conv}(\Pi(\hat{x})) \) by dualizing the constraints

\[
\begin{align*}
\lambda^T \left( x - \sum_\nu x_\nu \right) + \sum_\nu \zeta_\nu^T (T x_\nu + W y_\nu \alpha_\nu - q \alpha_\nu) - \sum_\nu \mu_\nu^T (R_\nu x_\nu + Q_\nu y_\nu \alpha_\nu + s_\nu \alpha_\nu) \\
+ \sum_\nu \gamma_\nu^T (1 \alpha_\nu - x_\nu) + \sum_\nu \alpha_\nu - 1 &\geq 0 \\
\Rightarrow \\
\lambda^T x - \sum_\nu (\lambda^T - \zeta_\nu^T T + \mu_\nu^T R_\nu + \gamma_\nu^T) x_\nu \\
+ \sum_\nu (1 + \zeta_\nu^T W y_\nu - \zeta_\nu^T q - \mu_\nu^T Q_\nu y_\nu - \mu_\nu^T s_\nu + \gamma_\nu^T 1) \alpha_\nu - 1 &\geq 0
\end{align*}
\]

The above Lagrangian inequality indicates that any inequality of the form

\[
\lambda^T x - 1 \geq 0
\]

(6.5)

is valid for \( \text{conv}(\Pi) \) if and only if the coefficients \( \lambda \) satisfy the following constraints:

\[
\begin{align*}
\lambda - T^T \zeta_\nu + R_\nu^T \mu_\nu + \gamma_\nu &\geq 0, \\
- \zeta_\nu^T W y_\nu + \zeta_\nu^T q + \mu_\nu^T Q_\nu y_\nu + \mu_\nu^T s_\nu - 1 - \gamma_\nu^T 1 &\geq 0, \\
\lambda &\in \mathbb{R}^l, \quad \mu_\nu, \gamma_\nu \geq 0 \quad \forall \nu \in \mathcal{L}(\hat{x}).
\end{align*}
\]  

(6.6)

In particular, we can choose an appropriate set of coefficients \( \lambda, \mu_\nu \) and \( \gamma_\nu \) by solving the following linear program:

\[
\begin{align*}
\min &\quad \lambda^T \hat{x} \\
\text{s.t.} &\quad \lambda - T^T \zeta_\nu + R_\nu^T \mu_\nu + \gamma_\nu \geq 0, \\
&\quad - \zeta_\nu^T W y_\nu + \zeta_\nu^T q + \mu_\nu^T Q_\nu y_\nu + \mu_\nu^T s_\nu - 1 - \gamma_\nu^T 1 \geq 0, \\
&\quad \lambda \in \mathbb{R}^l, \quad \mu_\nu, \gamma_\nu \geq 0 \quad \forall \nu \in \mathcal{L}(\hat{x}).
\end{align*}
\]  

(6.7)
6.1 Application of the linearize-branch-and-union algorithm to the second-stage scenario problem \[ (5.1) \]

**Corollary 6.1** Consider applying Modified-Algorithm\[2\] to solve the second-stage scenario problem \[ (5.1) \] with the scenario index \( \omega \) and \( \hat{x} \) to optimality. Each leaf node \( \nu \) of the branch-and-bound tree should have the following form of linear program:

\[
\min_y \ h^\omega y - W^\omega y - q^\omega = 0, \quad R^\omega \hat{x} + Q^\omega y + s^\omega \leq 0, \quad y \in \mathbb{R}^{l_2 + l_3}.
\]

(6.8)

for some coefficient matrix \( R^\omega, Q^\omega \) and \( s^\omega \) depending on \( \omega \) and \( \nu \).

**Remark 6.1** The key feature of (6.8) is that the linear inequalities are parametrized by the first-stage variable value \( \hat{x} \), which is the foundation of generating a parametric cut for the first-stage problem that will be discussed later.

Consider the problem \( (5.1) \). When a first-stage solution \( \hat{x} \) is given, we then solve the (second-stage) scenario sub-problem Sub(\( \hat{x}, \omega \)) using the Modified-Algorithm\[2\]. Applying Corollary 6.1 to each scenario sub-problem, we can get the following LP formulation of each node \( \nu \in \mathcal{L}^\omega(\hat{x}) \):

\[
\min_y \ h^\omega y - W^\omega y - q^\omega = 0, \quad R^\omega \hat{x} + Q^\omega y + s^\omega \leq 0, \quad y \in \mathbb{R}^{l_2 + l_3}.
\]

(6.9)

where \( \mathcal{L}^\omega(\hat{x}) \) is the set of nodes of the branch-and-bound tree of the scenario \( \omega \) sub-problem Sub(\( \hat{x}, \omega \)), and \( \nu \) is the index of a node in \( \mathcal{L}^\omega(\hat{x}) \).

**Proposition 6.2** Let \( \hat{p}^\omega = \{ [x, y^\omega] : \ g_i(x, y^\omega) \leq 0 \ \forall i \in I, \ y^\omega \in \mathcal{Y} \} \). For a feasible first-stage solution \( \hat{x} \), let \( \hat{P}_\nu^\omega(\hat{x}) = \{ [x, y^\omega] : R^\omega \hat{x} + Q^\omega y^\omega + s^\omega \leq 0 \} \) for every \( \nu \in \mathcal{L}^\omega(\hat{x}) \). Then \( \hat{p}^\omega \subseteq \bigcup_{\nu \in \mathcal{L}^\omega(\hat{x})} \hat{P}_\nu^\omega(\hat{x}) \).

**Remark 6.2** We make some clarification on the notations used in this proposition: The polytope \( \hat{P}_\nu^\omega(\hat{x}) \) defined in the \( [x, y^\omega] \)-space is indexed by \( \hat{x} \) as the coefficient matrices \( R^\omega, Q^\omega \) and \( s^\omega \) are generated based on an first-stage solution \( \hat{x} \). The point of the proposition is to show that the set \( \hat{p}^\omega \) which is independent of \( \hat{x} \) is contained in the union of polytopes that are defined using inequalities having \( \hat{x} \)-dependent coefficients.

**Proof** Based on the proof of Lemma 6.1, the inequalities \( R^\omega \hat{x} + Q^\omega y^\omega + s^\omega \leq 0 \) can be partitioned into two groups of inequalities: Group 1, the valid inequalities of the form \( C(k)^\omega (x, y^\omega) \leq 0 \) and \( D(k)^\omega (x, y^\omega) \leq 0 \) in the \( [x, y^\omega] \)-space, and Group 2, the simple-bound inequalities of the form \( y_i \leq b \) and \( y_i \geq b \) in the \( y^\omega \)-space. Theorem 4.1 indicates that the Group 1 of inequalities are valid for \( \hat{p}^\omega \). Denote the feasible regions induced by the Group 1 and Group 2 of constraints as \( P_1 \) (independent of node) and \( P_2^\omega \) (dependent on the node). Clearly, we have

\[
\hat{p}^\omega \subseteq P_1, \quad \{ [x, y^\omega] : y^\omega \in \mathcal{Y} \} \subseteq \bigcup_{\nu \in \mathcal{L}^\omega(\hat{x})} P_2^\omega.
\]
Therefore,
\[ \vartheta^\omega \subseteq \bigcup_{\nu \in L^\omega (\hat{x})} (P^1 \cap P^2_\nu) = \bigcup_{\nu \in L^\omega (\hat{x})} P^\omega_\nu (\hat{x}). \]

This concludes the proof.

Suppose \( x^* \) is an optimal first-stage solution of (TSS-MICP) and \( y^{\omega*} \) is an optimal solution of \( \text{Sub}(x^*, \omega) \). Since \( [x^*, y^{\omega*}] \in \vartheta^\omega \), Proposition 6.2 indicates that \( \exists \nu^* \in L^\omega (\hat{x}) \) such that \( [x^*, y^{\omega*}] \in P^\omega_{\nu^*} (\hat{x}) \). The LP relaxation problem at a node \( \nu \in L^\omega (\hat{x}) \) is

\[
\begin{aligned}
\min_{y^\omega_\nu} & \quad h^\omega_\nu^T y^\omega_\nu \\
\text{s.t.} & \quad W^\omega_\nu y^\omega_\nu = q^\omega - T^\omega_\nu \hat{x}, \quad \text{dual } \theta^\omega_\nu \\
& \quad Q^\omega_\nu y^\omega_\nu \leq -R^\omega_\nu \hat{x} - s^\omega_\nu, \quad \text{dual } \mu^\omega_\nu \\
& \quad y^\omega_\nu \in \mathbb{R}^{l_2 + l_3},
\end{aligned}
\]

where \( \theta^\omega_\nu \) and \( \mu^\omega_\nu \geq 0 \) is the optimal value of the dual vector obtained after solving the node LP relaxation problem. The optimal objective of (6.1) can be represented using the dual vector as

\[
\theta^\omega_\nu^T x = (\hat{x}^T q^\omega - \theta^\omega_\nu^T T^\omega_\nu x) + \mu^\omega_\nu^T (R^\omega_\nu \hat{x} + s^\omega_\nu).
\]

Define the polyhedron \( V^\omega_\nu (\hat{x}) \) as follows:

\[
V^\omega_\nu (\hat{x}) := \left\{ \eta^\omega, x \middle| \eta^\omega = \begin{cases} \eta^\omega_\nu & \nu \in L^\omega (\hat{x}) \\ \eta^\omega & \nu \notin L^\omega (\hat{x}) \end{cases}, x = \begin{cases} x_\nu & \nu \in L^\omega (\hat{x}) \\ x & \nu \notin L^\omega (\hat{x}) \end{cases}, \right. \quad (6.11)
\]

where \( \eta^\omega \) here represent a \( l_1 \) dimensional vector with every entry being 1. We must have \( [x^*, \eta^{\omega*}] \in V^\omega_\nu (\hat{x}) \). Let \( \Pi^\omega (\hat{x}) = \bigcup_{\nu \in L^\omega (\hat{x})} V^\omega_\nu (\hat{x}) \). The convex hull \( \text{conv}(\Pi^\omega (\hat{x})) \) can be represented following [8,9] as

\[
\text{conv}(\Pi^\omega (\hat{x})) = \text{Proj}_{\eta^\omega, x, \sum_{\nu \in L^\omega (\hat{x})} \alpha_\nu} \left\{ \eta^\omega, x, \eta^\omega_\nu, x_\nu, \alpha_\nu \middle| \begin{array}{l}
\eta^\omega = \sum_{\nu \in L^\omega (\hat{x})} \eta^\omega_\nu \\
x = \sum_{\nu \in L^\omega (\hat{x})} x_\nu \\
\eta^\omega_\nu \geq (\mu^\omega_\nu^T R^\omega_\nu - \theta^\omega_\nu^T T^\omega_\nu) x_\nu + (\mu^\omega_\nu^T s^\omega_\nu + \theta^\omega_\nu^T q^\omega) \alpha_\nu \\
Ax_\nu \leq b \alpha_\nu, \quad 0 \leq x_\nu \leq 1 \alpha_\nu \\
\sum_{\nu \in L^\omega (\hat{x})} \alpha_\nu = 1, \quad \alpha_\nu \geq 0 \\
\forall \nu \in L^\omega (\hat{x}) \end{array} \right. \quad (6.12)
\]

**Observation 1** Any feasible \( [x, \eta^\omega] \) of (TSS-MICP) is in \( \text{conv}(\Pi^\omega (\hat{x})) \), where \( \eta^\omega \) is an auxiliary variable representing the optimal value of (6.1).
We can parameterize all the valid inequalities for \( \text{conv}(\Pi^\omega) \) by dualizing the constraints

\[
\eta^\omega - \sum_{\nu} \eta_{\nu}^\omega + \lambda^T \left( \sum_{\nu} x_{\nu} - x \right) + \sum_{\nu} \left( \eta_{\nu}^\omega - \mu_{\nu}^\omega R_{\nu}^\omega x_{\nu} + \theta_{\nu}^\omega T^\omega x_{\nu} - \mu_{\nu}^\omega s_{\nu}^\omega \alpha_{\nu} - \theta_{\nu}^\omega q_{\nu}^\omega \alpha_{\nu} \right) \\
- \beta_{\nu}^T (b_{\nu} - A x_{\nu}) + \sum_{\nu} \gamma_{\nu}^T (1 \alpha_{\nu} - x_{\nu}) + \sigma \left( \sum_{\nu} \alpha_{\nu} - 1 \right) \geq 0
\]

\[
\Rightarrow \eta^\omega - \lambda x + \sum_{\nu} \left( \lambda^T - \mu_{\nu}^\omega R_{\nu}^\omega + \theta_{\nu}^\omega T^\omega - \beta_{\nu}^T A - \gamma_{\nu}^T \right) x_{\nu} \\
+ \sum_{\nu} \left( \sigma - \mu_{\nu}^\omega s_{\nu}^\omega - \theta_{\nu}^\omega q_{\nu}^\omega + \beta_{\nu}^T b + \gamma_{\nu}^T \right) \alpha_{\nu} - \sigma \geq 0
\]

The above Lagrangian inequality indicates that any inequality of the form

\[
\eta^\omega \geq \lambda^T x + \sigma
\]

is valid for \( \text{conv}(\Pi^\omega) \) if and only if the coefficients \( \lambda \) and \( \sigma \) satisfy the following constraints:

\[
\begin{align*}
\lambda &- R_{\nu}^\omega \mu_{\nu}^\omega + \theta_{\nu}^\omega T^\omega - A^T \beta_{\nu} - \gamma_{\nu} \leq 0, \\
\sigma &- \mu_{\nu}^\omega s_{\nu}^\omega - \theta_{\nu}^\omega q_{\nu}^\omega + \beta_{\nu}^T b + \gamma_{\nu}^T \leq 0, \\
\lambda &\in \mathbb{R}^{l_1}, \sigma \in \mathbb{R}, \beta_{\nu}, \gamma_{\nu} \geq 0 \quad \forall \nu \in \mathcal{L}^\omega(\hat{x}).
\end{align*}
\]

In particular, we can choose an appropriate set of coefficients \( \lambda \) and \( \sigma \) by solving the following linear program:

\[
\begin{align*}
\text{max} & \quad \lambda^T \hat{x} + \sigma \\
\text{s.t.} & \quad \lambda - R_{\nu}^\omega \mu_{\nu}^\omega + \theta_{\nu}^\omega T^\omega - A^T \beta_{\nu} - \gamma_{\nu} \leq 0, \\
& \quad \sigma - \mu_{\nu}^\omega s_{\nu}^\omega - \theta_{\nu}^\omega q_{\nu}^\omega + \beta_{\nu}^T b + \gamma_{\nu}^T \leq 0, \\
& \quad \lambda \in \mathbb{R}^{l_1}, \sigma \in \mathbb{R}, \beta_{\nu}, \gamma_{\nu} \geq 0 \quad \forall \nu \in \mathcal{L}^\omega(\hat{x}).
\end{align*}
\]

**Lemma 6.2** Let \( \hat{x} \) be any first-stage feasible solution of \( \text{TSS-MICP} \) and \( \omega \in \Omega \) be a scenario. Suppose the second-stage problem \( \text{Sub}(\hat{x}, \omega) \), i.e., the problem \( \text{(5.1)} \) with \( x = \hat{x} \) has been solved to optimality using a linearize-and-branch algorithm, and \( \mathcal{L}^\omega(\hat{x}) \) be the set of leaf nodes of the branch-and-bound tree available at termination of the linearize-and-branch algorithm. Then we have:

\[
\begin{align*}
Q(\hat{x}, \xi^\omega) &= \lambda^{\omega*} \hat{x} + \sigma^{\omega*}, \\
Q(x, \xi^\omega) &\geq \lambda^{\omega*} x + \sigma^{\omega*},
\end{align*}
\]

for all feasible first-stage solution \( x \), where the coefficients \( \lambda^{\omega*} \) and \( \sigma^{\omega*} \) are the optimal solution of \( \text{(6.15)} \).

**Proof** The proof is similar to the proof of Lemma 2.1 in [21].
Lemma 6.2 indicates that we can aggregate the scenario valid inequality \( \eta \geq \lambda^\omega^* x + \sigma^\omega^* \) with the weight given by the worst-case probability distribution to generate a valid inequality to add to the first-stage master problem. With this property, we can then establish a decomposition linearize-branch-and-union based algorithm (Algorithm 5) for solving \((TSS-MICP)\). The idea is similar to Algorithm 4. The difference is on the approach of solving the scenario sub-problems to generate a valid scenario cut. In Algorithm 4 the scenario sub-problem is solved using a pure parametric cutting-plane algorithm and the scenario cut is generated by taking the dual of the eventual LP relaxation. While in Algorithm 5 it is solved using the linearize-and-branch algorithm, and the scenario cut is generated using a polytope-union technique. Together we call it decomposition algorithm with linearize-branch-and-union technique.

In particular, the master problem at iteration \( m \) of Algorithm 5 has the following formulation:

\[
\begin{align*}
\min_x & \quad c^\top x + \eta \\
\text{s.t.} & \quad Ax \leq b, \\
& \quad \eta \geq \sum_{\omega \in \Omega} p^k\lambda^{k\omega^*} x + \sum_{\omega \in \Omega} p^k\sigma^{k\omega^*} \quad \forall k \in [m - 1], \\
& \quad x \in \mathcal{C} \cap \{0, 1\}^l,
\end{align*}
\tag{6.17}
\]

where \( \eta \geq \sum_{\omega \in \Omega} p^k\lambda^{k\omega^*} x + \sum_{\omega \in \Omega} p^k\sigma^{k\omega^*} \) is the (aggregated) first-stage valid inequality generated at the end of iteration \( k \) and \( p^k = \{ p^{k\omega} | \omega \in \Omega \} \) is an optimal solution of \((5.3)\) with \( \hat{x} \leftarrow x^k \), and \( x^k \) is an optimal solution of the master problem \((6.17)\) at the beginning of iteration \( k \).

**Theorem 6.1** Let Assumption 4 hold. If the master problem and scenario sub-problems are solved to optimality in every iteration of Algorithm 5, the algorithm can return an optimal solution of \((TSS-MICP)\) in a finite number of iterations.

**Proof** The proof is similar to the proof of Theorem 4.1 in [21].

7 Numerical Experimentation

We evaluated the performance of the linearize-branch-and-union algorithm using the following distributionally robust two-stage stochastic mixed-integer convex programming problem. A Wasserstein-distance based ambiguity set \( \mathcal{P}^W \) with finite support is used in the distributional robustness formulation. Our test model take the form:


**Algorithm 5** A decomposition linearize-branch-and-union based algorithm for solving (TSS-MICP).

1: Initialization: $L \leftarrow -\infty$, $U \leftarrow \infty$, $m \leftarrow 1$.
2: while $U - L > 0$ do
3: Solve the first-stage master problem (6.17) to optimality with main iteration index being $m$.
4: Let $(\eta^m, x^m)$ be the optimal solution.
5: Update the lower bound as $L \leftarrow c^\top x^m + \eta^m$.
6: Set the current best solution as $x^* \leftarrow x^m$.
7: for $\omega \in \Omega$ do
8: Solve the second-stage problem Sub($x^m$, $\omega$) to optimality using the Modified-
9: Algorithm 2.
10: When the Modified-Algorithm 2 terminates, it generates a branch-and-bound tree with the set $L_\omega(x^m)$ of nodes.
11: Solve (6.15) with $\hat{x} \leftarrow x^m$ to determine the coefficients $\lambda^{m, \omega*}$ and $\sigma^{m, \omega*}$.
12: Obtain the worst-case probability distribution $p^m = \{p^{m, \omega} \mid \omega \in \Omega\}$ by solving (5.3) with $\hat{x} \leftarrow x^m$.
13: Generate the following aggregated Benders’ cut with iteration index $m$:
14: $\eta \geq \sum_{\omega \in \Omega} p^m \lambda^{m, \omega*} y^\omega + \sum_{\omega \in \Omega} p^m \sigma^{m, \omega*}$.
15: Add the Benders’ cut to the first-stage master problem (6.17).
16: Update the upper bound as $U \leftarrow \min \{U, c^\top x^m + \sum_{\omega \in \Omega} p^m Q(x^m, \xi^\omega)\}$.
17: $m \leftarrow m + 1$.
18: end while
19: Return $x^*$.

\[
\min_x c^\top x + \max_{p \in W} \mathbb{E}[Q(x, \xi)] \quad (7.1)
\]
\[
s.t. \quad Fx \geq a \quad (7.2)
\]
\[
x \in \{0, 1\}^{l_0} \quad (7.3)
\]

where, $Q(x, \xi^\omega) = \min_{y^\omega} d^\omega y^\omega$\quad (7.4)
\[
s.t. \quad B^\omega y^\omega \leq b^\omega \quad (7.5)
\]
\[
D^1 y^\omega - \log(1 + \exp\{D^2 y^\omega\}) \geq f^\omega + Tx \quad (7.6)
\]
\[
y^\omega \in \mathbb{R}^{l_1} \times 2^{l_2} \quad (7.7)
\]

and
\[
W = \left\{ \begin{array}{c}
\exists q \in \mathbb{R}^{(|\Omega|)}: \\
\sum_{i=1}^{l_1} \sum_{j=1}^{l_2} \|\xi_i - \xi_j\| q_{ij} \leq \rho \\
\sum_{i=1}^{l_1} q_{ij} = p^0_j \quad \forall j \in \Omega \\
\sum_{j=1}^{l_2} q_{ij} = p_i \quad \forall i \in \Omega \\
p^0_i \geq 0 \quad \forall i \in [|\Omega|]
\end{array} \right\}, \quad (Wass)
\]

where $p^0$ is a reference probability distribution and $q$ is a joint distribution over the finite support, $\rho$ is a given radius, and $\xi_i$ represents a realization of random vector $\xi$ for scenario $i \in \Omega$. 

7.1 Data Generation

We consider the above model in four different sizes (denoted as S1, S2, S3, and S4) in our experiments. The first-stage has only one constraint in all the generated problems. We denote the number of first-stage variable by \( l_0 \).

Let the cardinality of the second-stage linear and nonlinear constraints respectively be \( m_1 \) and \( m_2 \), and \( m = m_1 + m_2 \). We set \( m_1 = \lfloor 0.2m \rfloor \) (hence \( m_2 = m - m_1 \)) and \( l_2 = \lfloor \ell/3 \rfloor \) where \( \ell = l_1 + l_2 \). S1, S2, S3 and S4 problems are respectively specified by taking \( l_0 = 100, \ell = 30, m = 20, |\Omega| = 10; \)
\( l_0 = 100, \ell = 50, m = 40, |\Omega| = 50; \)
\( l_0 = 100, \ell = 150, m = 100, |\Omega| = 50 \) and \( l_0 = 100, \ell = 150, m = 100, |\Omega| = 200 \). For each problem size, we generate 10 different instances for which the parameters are generated randomly as follows. We choose first-stage objective coefficient \( c \) randomly from the interval \([0,1]\) using a uniform distribution. Left-hand side coefficients of the first-stage constraint are set to be all one and the right-hand side constant is set to be 0.4 times of the total number of first-stage variables. For each scenario \( \omega \in \Omega \), the second-stage objective and linear-constraint coefficients \( d^\omega, B^\omega \) are randomly picked from the interval \([50,70]\) and \([0.5,1.5]\), both following a uniform distribution. In second-stage non-linear constraints, these intervals for \( D^{1\omega} \) and \( D^{2\omega} \) respectively are \([5,10]\) and \([5,6]\). Finally, random data intervals used for technology matrix \( T \) generation and right hand side constant \( f^\omega \) of non-linear constraints respectively are \([90,95]\) and \([20,25]\). Note that the technology matrix is assumed to be indifferent over scenario \( \omega \).

7.2 Implementation Details

We implement two variants of the Branch-and-union (B&U) algorithm (Branch-union (no cut) and Branch-cut-union) to solve the second-stage scenario problem and subsequently generate aggregated optimality cut for the master problem. Branch-and-union (no cut) variant is a counterpart to the branch-and-union algorithm proposed in [21]. Instead of a mixed-integer conic programming problem, here we adapt it for a mixed-integer general convex programming problem. This variant solves the non-linear convex subproblems arising in the branch-and-bound (B&B) tree. Branching continues until desired optimality gap or maximum leaf node tolerance is achieved. Since leaf-node problems are (non-linear) convex optimization problems, tangent planes at the solution of these problems are generated to build leaf-node polytopes for the disjunction-based cut-generation linear programs (CGLP). The difference between these two implementation variants is that subproblems in the latter variant are linear while the former repeatedly solves convex subproblems. Note that the no-cut version does not require a refinement phase. We further discuss this comparative performance based on the following numerical findings.

The Branch-cut-union follows the structure of Modified-Algorithm 2 with a minor modification. Step-1 of Modified-Algorithm 2 starts with only the linear constraints of the second-stage scenario problems. The next step generates a
linear approximation of each of the convex constraints using the projection point generated from the projection problem in \( y \)-space while fixing \( x = x_m \), \( x_m \) to be the current first-stage solution. Thus, each projection problem in modified step-2 generates \( m_2 \) linear constraints. After incorporating these new linear constraints, Step-1 is re-executed. Looping between Step-1 and modified Step-2 (referred as the refinement phase) continues for ten times or until the difference between the objective values obtained from two consecutive step-1 reaches desired gap (\( 1 \times 10^{-6} \)). The resulting MILP, that contains linear cuts for the non-linear convex constraints, is then provided as input to B&B tree generation oracle for having the polytope information at the leaf nodes collected. Such leaf-node polytope information provides necessary inputs for disjunction-based CGLP. Termination criteria to stop branching in this B&B tree is same as that of the branch-and-union algorithm.

7.3 Experimental Setup

All the computations are performed on a 64-core server having Xeon 2.20 GHz CPUs and 128 GB RAM. For each instance, scenario problems are solved in parallel using one core for each scenario and 10 cores in total at the same time. Our implementation uses Python 3.7. The ‘multiprocessing’ package is used for assigning a scenario problem to a CPU core and thus solving the scenario problems in parallel, and the package ‘time’ is used to keep track of computation time. The Gurobi solver (‘gurobipy’ package) is used for solving mixed-integer linear programming master problems, the LP sub-problems arising in the B&B tree, cut-generation-linear-programs (CGLP) in each scenario problem, and the linear programs used to generate the worst case probability distributions from the finitely supported Wasserstein ambiguity set. In the branch-and-bound (B&B) type algorithm implementation, the data structure ‘PriorityQueue’ from the ‘queue’ package is used so that the leaf node having the least objective value can be easily accessed. The IPOPT solver (‘cyipopt’ package) is used to solve all the convex sub-problems and projection problems with a warm start, where possible.

7.4 Experimental Results

We report the computational findings in Table 1 respectively for problems S1, S2, and S3. We provide the time required for solving the two-stage DRO problem when Branch-cut-union and Branch-union (no-cut) frameworks are used to solve the second-stage scenario problem. We also provide the number of master iterations required by both approaches in achieving relative optimality gap of \( 1 \times 10^{-3} \). Additionally, for both variants, the lower bound and upper bounds obtained by the proposed decomposition algorithm at termination are provided. Furthermore, the total number of leaf-node polytopes used for CGLP over all master iterations is reported. This number is same for both
variants, hence, it is provided only once. This is because (i) polytope given to B&B tree generation oracle under branch-cut-union approach is a ‘reasonably tight approximation’ of the non-linear constraints, (ii) leaf-node count-based stopping criterion remains the same for both. Additionally, we observe that the total number of cuts generated at termination in a cut-based approach is same when the instance requires the same number of master iterations. Note that we are adding all the constraints in each refinement phase using a projection point and the number of refinement phase required by different instances appears to be the same for these instances.

| Problem Size: S1 (l₀ = 100, ℓ = 30, m = 20, |Γ| = 10) | Linearize-branch-and-union | Branch-and-union (No linearization) |
|---|---|---|---|---|---|---|
| Ins | Ms It# | Time (s) | LB, UB | Leaf Nodes | Cuts | Ms It# | Time (s) | LB, UB |
| 1 | 2 | 10.44 | 655.27, 655.27 | 167 | 320 | 2 | 25.40 | 655.27, 655.27 |
| 2 | 2 | 9.44 | 653.94, 653.94 | 137 | 320 | 2 | 20.29 | 653.94, 653.94 |
| 3 | 2 | 18.73 | 643.56, 643.56 | 133 | 320 | 2 | 20.99 | 643.56, 643.56 |
| 4 | 2 | 9.88 | 633.64, 633.64 | 125 | 320 | 2 | 18.38 | 633.64, 633.64 |
| 5 | 2 | 24.55 | 622.34, 622.94 | 297 | 960 | 6 | 43.28 | 622.34, 622.94 |
| 6 | 2 | 11.38 | 639.19, 639.19 | 146 | 320 | 2 | 20.35 | 639.19, 639.19 |
| 7 | 6 | 31.65 | 639.5, 634.43 | 499 | 960 | 6 | 75.57 | 639.5, 634.43 |
| 8 | 2 | 9.56 | 666.01, 666.01 | 136 | 320 | 2 | 20.84 | 666.01, 666.01 |
| 9 | 2 | 10.19 | 631.71, 631.71 | 105 | 320 | 2 | 15.80 | 631.71, 631.71 |
| 10 | 2 | 10.01 | 626.66, 626.66 | 155 | 320 | 2 | 22.77 | 626.66, 626.66 |

| Problem Size: S2 (l₀ = 100, ℓ = 50, m = 40, |Γ| = 50) | Linearize-branch-and-union | Branch-and-union (No linearization) |
|---|---|---|---|---|---|---|
| Ins | Ms It# | Time (s) | LB, UB | Leaf Nodes | Cuts | Ms It# | Time (s) | LB, UB |
| 1 | 2 | 136.31 | 660.39, 660.39 | 1482 | 3200 | 2 | 425.99 | 660.39, 660.39 |
| 2 | 2 | 128.44 | 668.35, 668.35 | 1494 | 3200 | 2 | 463.67 | 668.35, 668.35 |
| 3 | 2 | 152.94 | 669.94, 669.94 | 1570 | 3200 | 2 | 492.64 | 669.94, 669.94 |
| 4 | 2 | 125.84 | 657.66, 657.66 | 1210 | 3200 | 2 | 576.26 | 657.66, 657.66 |
| 5 | 2 | 138.01 | 691.5, 670.15 | 1573 | 3200 | 2 | 497.46 | 670.15, 670.15 |
| 6 | 2 | 132.45 | 608.91, 608.91 | 1618 | 3200 | 2 | 484.60 | 608.91, 608.91 |
| 7 | 2 | 121.48 | 671.55, 671.55 | 1453 | 3200 | 2 | 473.04 | 671.55, 671.55 |
| 8 | 2 | 140.19 | 665.93, 665.93 | 1455 | 3200 | 2 | 471.84 | 665.93, 665.93 |
| 9 | 2 | 133.02 | 668.91, 668.91 | 1525 | 3200 | 2 | 492.44 | 668.91, 668.91 |
| 10 | 2 | 114.48 | 667.97, 667.97 | 1314 | 3200 | 2 | 394.26 | 667.97, 667.97 |

| Problem Size: S3 (l₀ = 100, ℓ = 150, m = 100, |Γ| = 50) | Linearize-branch-and-union | Branch-and-union (No linearization) |
|---|---|---|---|---|---|---|
| Ins | Ms It# | Time (s) | LB, UB | Leaf Nodes | Cuts | Ms It# | Time (s) | LB, UB |
| 1 | 2 | 377.03 | 667.77, 667.77 | 7058 | 8000 | 2 | 21584.71 | 667.77, 667.77 |
| 2 | 2 | 3120.86 | 664.49, 664.49 | 5724 | 8000 | 2 | 20804.22 | 664.49, 664.49 |
| 3 | 2 | 4710.01 | 671.43, 671.43 | 8368 | 8000 | 2 | 30369.63 | 671.43, 671.43 |
| 4 | 2 | 3464.19 | 666.51, 666.51 | 6388 | 8000 | 2 | 22881.88 | 666.51, 666.51 |
| 5 | 2 | 3736.66 | 672.90, 672.90 | 7185 | 8000 | 2 | 24830.48 | 672.90, 672.90 |
| 6 | 2 | 2887.97 | 666.23, 666.23 | 5403 | 8000 | 2 | 19586.64 | 666.23, 666.23 |
| 7 | 2 | 3013.14 | 674.11, 674.11 | 6980 | 8000 | 2 | 24943.91 | 674.11, 674.11 |
| 8 | 2 | 2901.01 | 667.37, 667.37 | 5900 | 8000 | 2 | 21042.43 | 667.37, 667.37 |
| 9 | 2 | 3263.84 | 668.54, 668.54 | 6143 | 8000 | 2 | 21980.14 | 668.54, 668.54 |
| 10 | 2 | 3401.53 | 666.76, 666.76 | 6698 | 8000 | 2 | 24236.66 | 666.76, 666.76 |
Table 2: Experimental findings for problem size S4 with 200 scenarios. Leaf nodes and cuts shown are aggregated over all scenarios and master iterations (Ms. It#)

| Ins | Ms It# | Time (s)  | LB, UB   | Leaf Nodes | Cuts | Ms It# | Time (s)  | LB, UB   |
|-----|--------|-----------|----------|------------|------|--------|-----------|----------|
| 1   | 2      | 7605.49   | 667.92, 667.92 | 27983      | 32000 | 2      | 98207.42  | 667.92, 667.92 |
| 2   | 2      | 7707.16   | 664.84, 665.84 | 28088      | 32000 | 2      | 102952.85 | 665.84, 665.85 |
| 3   | 2      | 7054.60   | 668.51, 668.58 | 27066      | 32000 | 2      | 98983.28  | 668.51, 668.58 |
| 4   | 2      | 8285.40   | 669.31, 669.31 | 27146      | 32000 | 2      | 101576.39 | 669.31, 669.31 |
| 5   | 2      | 7539.02   | 671.27, 671.27 | 28249      | 32000 | 2      | 104929.77 | 671.27, 671.27 |
| 6   | 2      | 7001.11   | 665.77, 665.77 | 24997      | 32000 | 2      | 93984.88  | 665.77, 665.77 |
| 7   | 2      | 8766.54   | 671.24, 671.24 | 28410      | 32000 | 2      | 104406.79 | 671.24, 671.24 |
| 8   | 2      | 7539.79   | 668.73, 668.73 | 25856      | 32000 | 2      | 95901.45  | 668.73, 668.73 |
| 9   | 2      | 6821.77   | 666.84, 666.84 | 23034      | 32000 | 2      | 85558.73  | 666.84, 666.84 |
| 10  | 2      | 8895.12   | 667.64, 667.64 | 30045      | 32000 | 2      | 110624.32 | 667.64, 667.64 |

In terms of computational time, the branch-cut-union approach outperforms the non-cut variant. On average, on the smaller instances the computational time is reduced by about 50%. However, as the problem size increases, this difference is much more significant. Problems in S2 and S3 are solved three to seven times faster. This difference is attributed to the time required to solve linear subproblems over non-linear subproblems in exploring the B&B tree nodes.

For problem size S4, we reconsider problem size S3 but with 200 scenarios. We report the associated findings in Table 2. Like other problem sizes we report the total cuts generated and leaf nodes used for CGLP aggregated over all scenarios and master iterations. It is observed that linearize-branch-union based decomposition approach is able to solve all the instances in approximately 2 hours on average while the other variant takes more than 24+ hours. In fact, it is able to complete only one master iteration if 24-hour time limit is given. Observe that although we do not increase the variable number for second stage, time taken by branch-and-union version (no linearization) is more than four fold compared to those in problem size S3.

A A Numerical Example of the Cutting-Plane Algorithm

Let us consider the following TSS-MICP:

\[
\begin{align*}
\min_{x_1, x_2 \in \{0,1\}} & \quad x_1 + 2x_2 + E[Q(x, \xi)] \\
\text{s.t.} & \quad 3x_1 + x_2 \geq 2,
\end{align*}
\]

(A.1)

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

\[
Q(x, \xi^{\omega_i}) = \min_{y_{11}, y_{12} \in \mathbb{Z}_+} 0.5y_{11} + y_{12}
\]

s.t. \[-2y_{11} - \log(1 + e) |y_{12} + \log(1 + e^{y_{11} + y_{12}}) \leq x_1 + x_2 - 1.\]

(A.2)
Solving the above updated master problem gives the first-stage solution problem gives: Benders’ cut is algorithm and the convergence results. These concepts are given in Definition B.1 to B.2.

The master problem at Iteration 1 is simply

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 (A.2) and solving the continuous relaxation sub-problem yields a fractional solution \([y_{11}, y_{12}] = [0.5, 0.48]\) which is on the curve \(-2y_{11} - \log(1 + e)(y_{11} + y_{12}) + \log(1 + e^{y_{11} + y_{12}}) = 0\). We add the tangent cut \(1.272y_{11} + 0.586y_{12} \geq 0.918\) induced by this fractional solution to the relaxation sub-problem. Then we solve the following relaxed MILP:

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 (A.2) for the given first-stage value \([x_1, x_2] = [1, 0]\).

Similarly, adding the tangent cut \(y_{11} - y_{12} + 1 \leq 0\), \(y_{11}, y_{12} \in \mathbb{Z}_+\),

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 (A.2) is:

The Benders’ cut associated with this scenario problem is \(\eta^{x_1} \geq 1 - 0.5x_1 - 0.5x_2\). Similarly, the Benders’ cut associated with the scenario \(\omega_2\) problem is \(\eta^{x_2} \geq 2 - x_1 - x_2\). The aggregated Benders’ cut is \(\eta \geq 1.5 - 0.75x_1 - x_2\). Adding this aggregated Benders’ cut to the master problem gives:

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 \([x_1, x_2] = [1, 0]\).

**B Solving MICP, JMICP and TSS-MICP to a Desired Tolerance**

**B.1 Analysis for the MICP problem**

We have shown that the algorithms given in previous sections can solve the mixed integer convex and conic programs (MICP) to optimality by purely generating cutting planes in finite number of iterations. This result was established under the assumptions that oracles for solving the projection problem (2.6), the convex optimization problem (2.5), and the supporting inequalities from Proposition 2.2 are known. However, in practice we may only be able to solve (2.6) and (2.5) to a desired precision. Thus we now develop an algorithm that solves the mixed-integer convex program (B.5) with a linear objective to \(\epsilon\)-accuracy only. In fact, this allows us to simplify steps given in Algorithms 1. Some careful definitions of optimality, feasibility and stability in the error-engaged case are necessary to depict the algorithm and the convergence results. These concepts are given in Definition B.3 to B.5.
Definition B.1 Consider a convex optimization problem:
\[
\min f(x) \quad \text{s.t.} \quad Ax = b, \ x \in \mathcal{C},
\]  
\tag{B.1}
where \(Ax = b\) represents linear equality constraints, \(\mathcal{C}\) is a full dimensional closed set in \(\mathbb{R}^{l_1+l_2}\). For any \(\epsilon > 0\), define the set \(C'_\epsilon\) as
\[
C'_\epsilon = \{x \in \mathbb{R}^{l_1+l_2} \mid \exists z \in \mathcal{C} \text{ s.t. } \text{dist}(x,z) \leq \epsilon\}.
\]
A point \(x\) is an \((\epsilon,\epsilon')\)-accurate solution of \(\text{[B.1]}\), if \(\text{[B.1]}\) is feasible, \(x \in C'_\epsilon\), and \(\text{OPT} - \epsilon \leq f(x) \leq \text{OPT} + \epsilon\), where \(\text{OPT}\) is the optimal value of \(\text{[B.1]}\).

Definition B.2 Consider a general mixed-integer programming problem
\[
\min f(x) \quad \text{s.t.} \quad Ax = b, \ x \in \mathcal{C}, \ x \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2},
\]  
\tag{B.2}
where \(\mathcal{C}\) is a full dimensional closed set in \(\mathbb{R}^{l_1+l_2}\). Let \(\epsilon, \epsilon'\) and \(\lambda\) be any three positive real numbers. Let \(C'_\epsilon\) be defined the same as in Definition \(\text{[B.1]}\) and define \(C'_\epsilon\) as
\[
C'_\epsilon = \mathcal{C} \setminus \bigcup_{x \in \partial\mathcal{C}} B(x, \epsilon),
\]
where \(\partial\mathcal{C}\) is the boundary of \(\mathcal{C}\) and \(B(x, \epsilon) = \{z \in \mathbb{R}^{l_1+l_2} \mid \|x - z\| \leq \epsilon\}\).

(i) The problem \(\text{[B.2]}\) is called \(\epsilon\)-feasible (infeasible) if \(C'_\epsilon \cap \{Ax = b\} \cap (\mathbb{Z}^{l_1} \times \mathbb{R}^{l_2})\) is non-empty (empty).

(ii) The problem is \((\epsilon,\lambda)\)-stable if it is \(\epsilon\)-feasible and there exist \(x^1\) and \(x^2\) such that \(x^1\) is an optimal solution of
\[
\min f(x) \quad \text{s.t.} \quad Ax = b, \ x \in C'_\epsilon, \ x \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2},
\]  
\tag{B.3}
\(x^2\) is an optimal solution of
\[
\min f(x) \quad \text{s.t.} \quad Ax = b, \ x \in C'_\epsilon, \ x \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2},
\]  
\tag{B.4}
\(x^1 = x^2\) and \(\|x^1 - x^2\| \leq \lambda\).

(iii) A point \(x \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2}\) is an \((\epsilon,\epsilon',\lambda)\)-accurate solution of \(\text{[B.2]}\), if \(\text{[B.2]}\) is \((\epsilon',\lambda)\)-stable, \(x \in C'_\epsilon\), and \(\text{OPT} - \epsilon \leq f(x) \leq \text{OPT} + \epsilon\), where \(\text{OPT}\) is the optimal value of \(\text{[B.2]}\).

For a geometric interpretation, the set \(C'_\epsilon\) is an \(\epsilon\)-covering of the set \(\mathcal{C}\), and the set \(C'_\epsilon\) can be thought of an epsilon-shrinkage of the set \(\mathcal{C}\). We consider the following mixed-integer convex program with a linear objective function
\[
\min c^T x \quad \text{s.t.} \quad Ax = b, \ x \in \mathcal{C}, \ x \in \mathbb{Z}^{l_1} \times \mathbb{R}^{l_2}.
\]  
\tag{B.5}
We focus on analyzing the case that the convex-optimization oracle can only identify a solution within a tolerable accuracy of optimality and feasibility. We make the following assumptions about the oracle on the solution accuracy.

Assumption 5 For any \(\epsilon,\epsilon' > 0\), there exists an oracle that can find a \((\epsilon,\epsilon')\)-accurate solution of a general convex optimization problem \(\text{[B.1]}\) given that \(\text{[B.1]}\) is feasible.

Note that for simplicity and the sake of our focus, under Assumption 5 we do not consider the negligible error that can happen while satisfying the linear equality constraint within feasibility tolerance. Furthermore throughout the analysis, it is assumed that linear program can be solved exactly. By utilizing the oracle provided in Assumption 5, Algorithm 6 is developed for solving \(\text{[B.5]}\) to \(\epsilon\)-accuracy. The algorithm is a modification of Algorithm 1 by removing the polishing step (Step 3), as it is not required to find an exact optimal solution in this case. The algorithm is essentially an outer approximation algorithm, where at each iteration a mixed-integer program obtained from the previous outer approximations is solved. The outer approximation is tightened by solving a projection problem approximately. The projection problems are defined at the solutions of the mixed-integer linear programs.
Algorithm 6 An Algorithm for solving a mixed integer convex program to a given accuracy.

1: **Input:** accuracy control parameters $\epsilon_1, \epsilon_2 > 0$.
2: Set $n \leftarrow 1$, $flag \leftarrow 0$ and $Z^{(0)} \leftarrow \emptyset$.
3: while $flag = 0$ do
4:  (Start iteration $n$.)
5:   **Step 1:** Solve the following master problem to optimality and let $x^{(n)}$ be the solution
6:   if it is feasible
7:       \[\begin{align*}
    \min & \quad c^\top x \\
    \text{s.t.} & \quad Ax = b, \\
    & \quad (x^{(k)} - z^{(k)})^\top x \leq (x^{(k)} - z^{(k)})^\top z^{(k)} \quad \forall k \in [n - 1], \\
    & \quad x \in \mathbb{Z}^1 \times \mathbb{R}^2,
    \end{align*}\]
8:   where $(x^{(k)} - z^{(k)})^\top x \leq (x^{(k)} - z^{(k)})^\top z^{(k)}$ is the projection cut generated
9:   at iteration $k$.
10:  if $x^{(n)} \in C$ then
11:     Stop and return $x^{(n)}$ as an optimal solution of (B.5).
12:  else
13:     Continue.
14:  end if
15: **Step 2:** Solve (B.6) to $(\epsilon_1, \epsilon_1)$-accuracy. Let $z^{(n)}$ and $d^{(n)}$ be a $(\epsilon_1, \epsilon_1)$-accurate solution (Definition B.1) and objective of (B.6).
16:   if $d^{(n)} \leq \epsilon_2/2$ then,
17:     Stop and return $x^{(n)}$.
18:   else
19:     Add the separation cut $(x^{(n)} - z^{(n)})^\top x \leq (x^{(n)} - z^{(n)})^\top z^{(n)}$ to (B.6).
20: end if
21: \(n \leftarrow n + 1\).
22: end while

Lemma B.1 Let $C$ be a full dimensional closed convex set in $\mathbb{R}^m$ and $v$ be a point in $\mathbb{R}^m \setminus C$. Consider the projection problem
\[
\min \|x - v\| \quad \text{s.t.} \quad x \in C. \tag{B.7}
\]
Let $x^*$ and $d$ be the optimal solution and optimal value of (B.7), respectively. Let $x^*$ be a $(\epsilon, \epsilon)$-accurate solution of (B.7) (Definition B.1), with $\epsilon < \frac{\epsilon_1}{\sqrt{m}}$. The following properties hold:
(a) $\|x^* - x^\star\| \leq 2\sqrt{m} + \epsilon$.
(b) Let $u$ be a point satisfying $\|u - v\| \leq d - 3\epsilon$, then $(v - x^\star)^\top (u - x^\star) > 0$, i.e., $u$ violates the approximated supporting inequality $(v - x^\star)^\top (x - x^\star) \leq 0$ of $C$.
(c) Suppose $C$ is bounded and $D := \max_{y_1, y_2 \in C} \|y_1 - y_2\|$ is the diameter of $C$. Let $r_{\text{max}} = \max_{z \in C \setminus V^-} \text{dist}(z, V^+)$, where $V^{\pm} = \{x \in \mathbb{R}^{1+k} \mid (v - x^\star)^\top (x - x^\star) \leq (\geq 0)\}$, and we define $r_{\text{max}} = 0$ if $C \cap V^- = \emptyset$. Then $r_{\text{max}}$ can be bounded as
\[r_{\text{max}} \leq \frac{9(D + 2d\sqrt{m})}{d}.
\]
Remark B.1 Note that in general the approximated projection point $x^*$ can locate inside $C$, and in this case the $r_{\text{max}}$ represents the maximum violated depth of the cut $(v - x^\star)^\top (x - x^\star) \leq 0$. See Figure 1 for an illustration.
Proof (a) Note that \( x^* = \text{proj}_C v \) and \( d = \text{dist}(v, C) \). Since \( x^* \) is an \((\epsilon, \epsilon)\)-accurate solution, by definition we have \( \| x^* - v \| \leq d + \epsilon \) and \( \text{dist}(x^*, C) \leq \epsilon \). It follows that
\[
(d + \epsilon)^2 \geq \| x^* - v \|^2 = \| x^* - x^* + x^* - v \|^2 \\
= \| x^* - x^* \|^2 + \| x^* - v \|^2 + 2\langle x^* - x^*, x^* - v \rangle \\
= \| x^* - x^* \|^2 + d^2 + 2\langle x^* - z, x^* - v \rangle + 2\langle z - x^*, x^* - v \rangle,
\]
where \( z = \text{proj}_C x^* \). Note that \( \langle z - x^*, x^* - v \rangle \geq 0 \) due to convexity and \( \| x^* - z \| = \text{dist}(x^*, C) \leq \epsilon \). Therefore, we have
\[
(d + \epsilon)^2 \geq \| x^* - x^* \|^2 + d^2 - 2d\epsilon,
\]
which implies that \( \| x^* - x^* \| \leq 2\sqrt{\epsilon} + \epsilon \).

(b) Note that we have
\[
\langle v - x^* \rangle (u - x^*) = \langle v - x^*, u - x^* \rangle \\
= \langle v - x^*, v - x^* \rangle + \langle v - x^*, u - v \rangle \\
\geq \| v - x^* \|^2 - \| v - x^* \| \| u - v \| \\
\geq (d - \epsilon)^2 - (d + \epsilon) \| u - v \| \\
\geq (d - \epsilon)^2 - (d + \epsilon)(d - 3\epsilon) = 4\epsilon^2 > 0,
\]
where we use the triangle inequality \( \| \text{dist}(v, C) - \text{dist}(x^*, C) \| \leq \| \text{dist}(v, C) \| \leq \| \text{dist}(v, C) \| \leq \| \text{dist}(v, C) \| \) in the above derivation.

(c) Since \( (v - x^*)^\top (x - x^*) \leq 0 \) is a supporting valid inequality of \( C \), it follows that
\[
C \cap V^- \subseteq \left\{ x \in \mathbb{R}^{n+1} \mid (v - x^*)^\top (x - x^*) \leq 0, (v - x^*)^\top (x - x^*) \geq 0 \right\}.
\]
Note that if the plane \( (v - x^*)^\top (x - x^*) = 0 \) does not intersect with \( C \), then \( r_{\text{max}} = 0 \) by definition. Let us consider the case in which this plane intersects with \( C \). Suppose \( r_{\text{max}} = \text{dist}(y, z) \) where \( y \in \partial \text{bd}(C) \) and \( z = \text{proj}_{V^+} y \) (illustrated in Figure 1). Note that in this case we have \( x^*, y, z \in C \), which implies that \( \| x^* - z \| \leq D \) by the definition of \( D \). We also note that \( y \) is on the line that passes \( z \) and is perpendicular to the plane \( (v - x^*)^\top (x - x^*) = 0 \), i.e., \( y \in \{ x : x - z = t(v - x^*), t \in \mathbb{R} \} \). This line intersects with the exact projection plane \( (v - x^*)^\top (x - x^*) = 0 \) at the point
\[
x^* = z - \frac{(v - x^*, z - x^*)}{\langle v - x^*, v - x^* \rangle} (v - x^*).
\]
Since \( y \) can be written as a convex combination of points \( z \) and \( x^* \), we have
\[
r_{\text{max}} = \| y - z \| \leq \| x^* - z \| = \left\| \frac{(v - x^*, z - x^*)}{\langle v - x^*, v - x^* \rangle} (v - x^*) \right\| \\
\leq \frac{|(v - x^*, z - x^*)|}{\langle v - x^*, v - x^* \rangle} \left( \| v - x^* \| + \| x^* - x^* \| \right) \left( \| v - x^* \| + \| x^* - x^* \| \right)
\]
We can evaluate the inner product in the numerator as follows
\[
(v - x^*, z - x^*) = (v - x^*, z - x^*) + (x^* - x^*, z - x^*) \\
= (v - x^*, z - x^*) + (v - x^*, x^* - x^*) + (x^* - x^*, z - x^*) + (x^* - x^*, x^* - x^*) \\
= (v - x^*, x^* - x^*) + (x^* - x^*, z - x^*) + (x^* - x^*, x^* - x^*),
\]
where we have used the property \( (v - x^*, z - x^*) = 0 \) to get the last equality. Therefore, the following bound holds
\[
|(v - x^*, z - x^*)| \leq (\| v - x^* \| + \| x^* - x^* \|) \| x^* - x^* \| + D \| x^* - x^* \| + \| x^* - x^* \|^2 \\
= (\| v - x^* \|^2 + 2\| x^* - x^* \|^2 + D) \| x^* - x^* \|.
\]
We Claim that $\epsilon < \frac{d}{64}$ implies $\epsilon \leq \sqrt{d}\epsilon$, and $2\sqrt{d}\epsilon + \epsilon \leq \frac{d}{2}$. To prove the claim, we first verify the first inequality: $\sqrt{d}\epsilon > \sqrt{64}\epsilon^2 = 8\epsilon > \epsilon$, and verify the second inequality: $\frac{d}{2} - \epsilon > \frac{d}{2} - \frac{d}{64} = \frac{31d}{64} > \frac{d}{2} > \sqrt{d} \cdot \frac{6\sqrt{d}}{8} > 2\sqrt{d}\epsilon$.

Applying the results from Part (a), we have the following bound for $r_{\text{max}}$:

$$r_{\text{max}} \leq \frac{(d + D + 4\sqrt{d}\epsilon + 2\epsilon)(2\sqrt{d}\epsilon + \epsilon)(d + 2\sqrt{d}\epsilon + \epsilon)}{d(d - 2\sqrt{d}\epsilon - \epsilon)} \leq \frac{9(D + 2d)\sqrt{d}}{d},$$

where the two inequalities in the claim are used to obtain the second inequality in the above derivation.

**Theorem B.1** Suppose $C$ is a full dimensional closed and bounded convex set. Suppose $C$ is contained in a ball of diameter $D \geq 1$. Consider applying Algorithm 6 to solve (B.5). If we set $\epsilon_1 = \epsilon_2^2$ (the parameters $\epsilon_1$ and $\epsilon_2$ are used in Algorithm 6) and $\epsilon_2 < \frac{1}{64}\min\{1, D\}$, then the following properties hold:

(a) Algorithm 6 terminates after a finite number of iterations.
(b) If the modified Algorithm 6 returns infeasible status, then (B.5) is $(20D\sqrt{\epsilon_2})$-infeasible.
(c) If the modified Algorithm 6 returns a solution $x^{(m)}$, then $x^{(m)}$ is a $(20\lambda\sqrt{\epsilon_2}\|c\|, 20D\sqrt{\epsilon_2}, \lambda)$-accurate solution of (B.5) given that (B.5) is $(20D\sqrt{\epsilon_2}, \lambda)$-stable.

**Remark B.2** After rescaling the error parameter $\epsilon_2$, Part (c) of Theorem B.1 can be restate as: The returned solution is a $(\rho\lambda\epsilon, \epsilon, \lambda)$-accurate solution of (B.5) if (B.5) is $(\epsilon, \lambda)$-stable, where $\rho$ is a constant only depending on $\|c\|$. 

Fig. 1: Geometry associated with Lemma B.1. The equations of hyperplanes $l_0$ and $l_1$ are $(v - x^*)^T(x - x^*) = 0$ and $(v - x')^T(x - x^*) = 0$, respectively. Note that the approximated projection point $x^*$ is located inside $C$ in this example.
Proof  (a) We prove by contradiction that the modified Algorithm 5 must terminate after finite number of iterations. If not, let \( x^* \) be a limit point of \( \{x^{(n)}\}_{n=1}^{\infty} \), and let \( x^{(n_k)} \to x^* \) as \( k \to \infty \). Case 1: dist(\( x^* , C \)) \( < \epsilon_2/4 \). In this case, there exists a \( n_k \) such that dist(\( x^{(n_k)} , C \)) \( \leq \epsilon_2/4 \). Recall that by Line 15 of Algorithm 5 \( z^{(n_k)} \) and \( d^{(n_k)} \) are the \((\epsilon_1, \epsilon_1)\)-accurate solution and objective of (2.6) at the main iteration \( n_k \). It follows that

\[
d^{(n_k)} = \left\| x^{(n_k)} - z^{(n_k)} \right\| \leq \text{dist}(x^{(n_k)}, C) + \text{dist}(z^{(n_k)}, C) \leq \frac{\epsilon_2}{4} + \epsilon_1 \leq \frac{\epsilon_2}{2}.
\]

It implies that the algorithm should terminate by Line 16 of Algorithm 6, which leads to a contradiction. Case 2: dist(\( x^* , C \)) \( \geq \epsilon_2/4 \). In this case, there exist a \( n_k \) such that dist(\( x^{(n_k)} , C \)) \( \geq 3\epsilon_2/16 \), and \( \left\| x^{(n_k)} - z^{(n_k)} \right\| \leq \frac{\epsilon_2}{4} \) for all \( n_i, n_j \geq n_k \). We have

\[
d^{(n_k)} = \left\| x^{(n_k)} - z^{(n_k)} \right\| \geq \text{dist}(x^{(n_k)}, C) - \text{dist}(z^{(n_k)}, C) \geq \frac{3\epsilon_2}{16}.
\]

and hence

\[
d^{(n_k)} - 3\epsilon_1 \geq \frac{\epsilon_2}{16} - 4\epsilon_1 = \frac{\epsilon_2}{8} + \left( \frac{\epsilon_2}{16} - \epsilon_1 \right) \geq \frac{\epsilon_2}{8}.
\]

Therefore, we have

\[
\left\| x^{(n_i)} - z^{(n_k)} \right\| \leq \frac{\epsilon_2}{8} \leq d^{(n_k)} - 3\epsilon_1,
\]

for any \( n_i > n_k \). Note that the cutting plane \( (x^{(n_k)} - z^{(n_k)})^\top (x - z^{(n_k)}) \leq 0 \) is added to the master problem. By Lemma B.11(b), we have

\[
(x^{(n_k)} - z^{(n_k)})^\top (x^{(n_i)} - z^{(n_k)}) \geq 0 \quad (B.8)
\]

However, the cutting plane \( (x^{(n_k)} - z^{(n_k)})^\top (x - z^{(n_k)}) \leq 0 \) has been added to the master problem at iteration \( n_k \). The (B.8) indicates that this constraint is violated by the master problem solution \( x^{(n)} \), which is a contradiction.

(b) Suppose the modified algorithm terminates and returns infeasible status. It implies that the algorithm should terminate by Line 16 of Algorithm 6, and \( \text{dist}(x^{(n_k)}, C) \leq \epsilon_2/4 \) for every \( n_k \leq n - 1 \), otherwise the algorithm will return a solution according to the termination criteria at Line 16. Let \( d^{(k)} \) denote the exact optimal objective of the projection problem at the \( k \)-th iteration, we should have

\[
d^{(k)} = \text{dist}(x^{(k)}, C)
\]

\[
\geq \text{dist}(x^{(k)}, z^{(k)}) - \text{dist}(z^{(k)}, C)
\]

\[
= d^{(k)} - \text{dist}(z^{(k)}, C)
\]

\[
= \frac{\epsilon_2}{2} - \epsilon_1 > \frac{\epsilon_2}{4}.
\]

Define the following violation depth of cut for each added cutting plane \( (x^{(k)} - z^{(k)})^\top (x - z^{(k)}) = 0 \) as

\[
r^{(k)} = \max_{z \in C \cap V^{(k)}_+} \text{dist}(z, V^{(k)}_+) \quad \text{if } C \cap V^{(k)}_+ \neq \emptyset,
\]

\[
r^{(k)} = \max_{z \in C \cap V^{(k)}_-} \text{dist}(z, V^{(k)}_-) \quad \text{if } C \cap V^{(k)}_- = \emptyset,
\]

where \( V^{(k)}_+ = \{ x \in \mathbb{R}^{l_1 + l_2} \mid (x^{(k)} - z^{(k)})^\top (x - z^{(k)}) \leq 0 \} \) and \( V^{(k)}_- = \{ x \in \mathbb{R}^{l_1 + l_2} \mid (x^{(k)} - z^{(k)})^\top (x - z^{(k)}) \geq 0 \} \). By Lemma B.1(c), we have

\[
r^{(k)} \leq \frac{9(D + 2d^{(k)})\sqrt{d^{(k)}\epsilon_1}}{d^{(k)}} = 9\sqrt{\epsilon_1} \left( \frac{D}{\sqrt{d^{(k)}}} + 2\sqrt{d^{(k)}} \right).
\]

Using \( \frac{\epsilon_2}{2} \leq d^{(k)} \leq D \) from (B.9), it follows that

\[
\frac{D}{\sqrt{d^{(k)}}} + 2\sqrt{d^{(k)}} \leq \max \left\{ 3\sqrt{D}, \frac{2D}{\sqrt{\epsilon_2}} + \sqrt{\epsilon_2} \right\} = \frac{2D}{\sqrt{\epsilon_2}} + \sqrt{\epsilon_2}.
\]
where we use the assumption $\epsilon_2 \leq D/64$ to get the last equality. Therefore, we obtain the following inequality:

$$r(k) \leq 9\sqrt{T_1} \left(\frac{2D}{\sqrt{\epsilon_2^2}} + \sqrt{\epsilon_2^2}\right) = 18D\sqrt{\epsilon_2} + 9\epsilon_2 \sqrt{\epsilon_2} \leq 20D\sqrt{\epsilon_2},$$

where we use the setting $\epsilon_1 = \epsilon_2^2$ to get the equality. It implies that the set $C^{\delta -} \cap (Z^{l_1} \times \mathbb{R}^{l_2})$ where $\delta = 20D\sqrt{\epsilon_2}$ is contained in the feasible set of the master problem at each iteration. Since the master problem at iteration $n$ is infeasible, (B.3) is $(20D\sqrt{\epsilon_2})$-infeasible.

(c) If the algorithm terminates and returns a solution $x^{(m)}$, we know that

$$\text{dist}(x^{(m)}, C) \leq \text{dist}(x^{(m)}, z^{(m)}) + \text{dist}(z^{(m)}, C)$$

$$= d^{(m)} + \text{dist}(z^{(m)}, C)$$

$$\leq \frac{c^2}{2} + \epsilon_1 < 20D\sqrt{\epsilon_2}.$$

It implies that $x^{(m)} \in C^{20D\sqrt{\epsilon_2}}$. Let $P$ be the polytope obtained after relaxing the integral constraints of the master problem at the termination of the algorithm. In the proof of Part (b), it has been shown that $C^{20D\sqrt{\epsilon_2}} \subseteq P$. Consider the following optimal values:

$$c^\top x^{(m)} = \min c^\top x \text{ s.t. } x \in P \cap [Ax = b] \cap (Z^{l_1} \times \mathbb{R}^{l_2}),$$

$$\text{OPT}^- = \min c^\top x \text{ s.t. } x \in C^{20D\sqrt{\epsilon_2}} \cap [Ax = b] \cap (Z^{l_1} \times \mathbb{R}^{l_2}),$$

$$\text{OPT}^+ = \min c^\top x \text{ s.t. } x \in C^{20D\sqrt{\epsilon_2}} \cap [Ax = b] \cap (Z^{l_1} \times \mathbb{R}^{l_2}).$$

The following inequalities hold:

$$\text{OPT}^+ \leq c^\top x^{(m)} \leq \text{OPT}^-,$$

$$\text{OPT}^+ \leq \text{OPT} \leq \text{OPT}^-,$$

where OPT is the optimal objective of (B.5). By Definition B.2(ii) and the assumption of $(20D\sqrt{\epsilon_2}, \lambda)$-stability, there exist $x^+$ and $x^-$ satisfying

$$x^+ \in C^{20D\sqrt{\epsilon_2}} \cap [Ax = b] \cap (Z^{l_1} \times \mathbb{R}^{l_2}),$$

$$x^- \in C^{20D\sqrt{\epsilon_2}} \cap [Ax = b] \cap (Z^{l_1} \times \mathbb{R}^{l_2}),$$

$$c^\top x^+ = \text{OPT}^+,$$

$$c^\top x^- = \text{OPT}^-,$$

$$x^+_z = x^-_z = b \text{ for some } b \in Z^{l_1},$$

$$\|x^+ - x^-\| \leq 20\lambda D\sqrt{\epsilon_2}.$$

It follows that $\text{OPT}^- - \text{OPT}^+ \leq 20\lambda D\sqrt{\epsilon_2} \|c\|$, and consequently

$$|c^\top x^{(m)} - \text{OPT}| \leq \text{OPT}^- - \text{OPT}^+ \leq 20\lambda D\sqrt{\epsilon_2} \|c\|,$$

which concludes the proof.

### B.2 Analysis for the JMICP problem

**Definition B.3** A feasible solution $x$ of JMICP is $\epsilon$-optimal if $\text{Obj}(x) \leq \text{Opt} + \epsilon$, where $\text{Obj}(x)$ is the objective value of JMICP at $x$ and Opt is the optimal objective of JMICP.
**Assumption 6** For any \( \bar{x} \in \mathcal{X} \cap \{0,1\}^{l_1} \), the convex set \( \mathcal{S}(\bar{x}) = \{ y \in \mathbb{R}^{l_2+l_3} : g_i(\bar{x}, y) \leq 0 \forall i \in I, \ y \in \mathcal{Y} \} \) is in full dimension, and the parametric MICP

\[
\min h^\top y \quad \text{s.t.} \quad T\bar{x} + Wy = q, \ g_i(\bar{x}, y) \leq 0 \forall i \in I, \ y \in \mathcal{Y} \cap (\mathbb{Z}^{l_2} \times \mathbb{R}^{l_3})
\]
is \((\epsilon, \lambda)\)-stable for some \( \epsilon, \lambda > 0 \).

**Lemma B.2** Suppose the assumptions [3] and [4] hold. Let \( \bar{x} \in \mathcal{X} \cap \{0,1\}^{l_1} \). Suppose Algorithm 6 is applied to the following parametric MICP problem:

\[
\min h^\top y \quad \text{s.t.} \quad T\bar{x} + Wy = q, \ g_i(\bar{x}, y) \leq 0 \forall i \in I, \ y \in \mathcal{Y} \cap (\mathbb{Z}^{l_2} \times \mathbb{R}^{l_3})
\]  

(B.10)

for solving it to the \((\rho_1, e_1, \lambda)\)-accuracy for some sufficiently small parameters \( e_1 > 0 \) and some constant \( \rho > 0 \). (This is achievable due to Theorem B.1.) Suppose Algorithm 6 generates the following linear program at termination:

\[
\min h^\top y \quad \text{s.t.} \quad T\hat{x} + Wy = q, \ A\hat{x} + By + C \leq 0,
\]  

(B.11)

where \( A, B \) and \( C \) are some coefficient matrices. For a given \( x \), denote \( \mathcal{Q}(x) \) as the optimal objective of the following MICP:

\[
\min h^\top y \quad \text{s.t.} \quad Tx + Wy = q, \ g_i(x, y) \leq 0 \forall i \in I, \ y \in \mathcal{Y} \cap (\mathbb{Z}^{l_2} \times \mathbb{R}^{l_3}).
\]

(B.12)

Then there exists some constant \( \rho_1 > 0 \) such that the following two inequalities hold:

\[
\mathcal{Q}(x) \geq (\lambda^\top T + \mu^\top A)x + (\lambda^\top q + \mu^\top C) - \rho_1\epsilon_1 \quad \forall x \in \mathcal{X} \cap \{0,1\}^{l_1},
\]

(B.13)

\[
\mathcal{Q}(x) \leq (\lambda^\top T + \mu^\top A)\bar{x} + (\lambda^\top q + \mu^\top C) + \rho_1
\]

(B.14)

where \( \lambda \) and \( \mu \geq 0 \) is the optimal dual vector of equality and inequality constraints in [4.2] and \( \mathcal{Q}(\cdot) \) is defined in [4.2].

**Proof** According to the mechanism of the cutting-plane method, the set \( Ax + By + C \leq 0 \) of constraints can be partitioned into two groups denoted as \( A_1x + B_1y + C_1 \leq 0 \) and \( A_2x + B_2y + C_2 \leq 0 \), where the first group of constraints are projection cutting planes obtained by solving the projection problem to a certain accuracy, while the second group of constraints are valid inequalities generated following the oracle in [11] for solving a MILP with pure cutting planes.

Let \( \mathcal{S} \) be defined in [11]. Since the inequalities \( A_1x + B_1y + C_1 \leq 0 \) are generated by solving the projection problem to a certain accuracy. It follows that there exists some constant \( \rho_1 > 0 \) such that

\[
\mathcal{S}_{\rho_1, \epsilon_1} \subseteq \{ [x, y] : A_1x + B_1y + C_1 \leq 0 \}.
\]

Furthermore, the oracle in [11] generates the valid inequalities \( A_2x + B_2y + C_2 \leq 0 \) such that

\[
\text{conv} \{ [x, y] : Tx + Wy = q, \ A_1x + B_1y + C_1 \leq 0, \ y \in \mathbb{Z}^{l_2} \times \mathbb{R}^{l_3} \}
\]

\[
\subseteq \mathcal{S} \cap \{ [x, y] : Tx + Wy = q, \ A_1x + B_1y + C_1 \leq 0 \}.
\]

Let \( \mathcal{Q}(\hat{x}) \) be the approximated objective. The accuracy control implies that

\[
\mathcal{Q}(\hat{x}) + \rho_1 \geq \mathcal{Q}(\hat{x}) = \min y \ h^\top y \quad \text{s.t.} \quad T\hat{x} + Wy = q, \ A\hat{x} + By + C \leq 0
\]

(B.17)

\[
\geq \mathcal{Q}(\hat{x}) - \rho_1.
\]

The strong duality gives

\[
(\lambda^\top T + \mu^\top A)\hat{x} + \lambda^\top q + \mu^\top C = \min_y \ h^\top y \quad \text{s.t.} \quad T\hat{x} + Wy = q, \ A\hat{x} + By + C \leq 0.
\]

(B.18)
Combining (B.17) and (B.18) leads to (B.14). For any \( x \in X \cap \{0,1\}^l \), the Assumption \([6]\) about stability implies that
\[
Q(x) = \min_{y} h^T y \{ \text{s.t. } (x,y) \in S \cap \{Tx + Wy = q\}, y \in \mathbb{Z}^l \times \mathbb{R}^l \},
\]
\[
Q(x) + \lambda_1 \epsilon_1 \geq \min_{y} h^T y \{ \text{s.t. } (x,y) \in S^{opt} \cap \{Tx + Wy = q\}, y \in \mathbb{Z}^l \times \mathbb{R}^l \}. \tag{B.19}
\]
Using (B.15) and (B.19) leads to the following inequalities:
\[
Q(x) + \lambda_1 \epsilon_1 \geq \min_{y} h^T y \{ \text{s.t. } Tx + Wy = q, A_1 x + B_1 y + C_1 \leq 0, y \in \mathbb{Z}^l \times \mathbb{R}^l \} \geq \min_{y} h^T y \{ \text{s.t. } Tx + Wy = q, Ax + By + C \leq 0 \}
\]
\[
\geq (-\lambda^T T + \mu^T A)x + (\lambda^T q + \mu^T C), \tag{B.20}
\]
which proves (B.13).

**Theorem B.2.** Let Assumptions \([5]\) and \([6]\) hold. Then Algorithm \([7]\) can identify a \(K\)-optimal solution of \((JMCP)\) in a finite number of iterations, where \(K\) is a constant and \(\epsilon_1\) is the accuracy control parameter used in Algorithm \([7]\).

**Proof.** We first show that if the algorithm terminates, it will return a number-optimal solution of \((JMCP)\). Suppose the algorithm terminate at iteration \(m\), and it returns a solution \(x^m\). The upper bound \(U^m\) at the termination step is determined by the following equation
\[
U^m = c^T x^m + \bar{Q}(x^m) \geq c^T x^m + Q(x^m) - \rho_1 \epsilon_1. \tag{B.21}
\]
The lower bound \(L^m\) at the termination step is equal to the optimal value of the master problem:
\[
L^m = \min_{x \in X \cap \{0,1\}^l} c^T x + \eta \quad \text{s.t. } \eta \geq a^k x + b^k \quad \forall k \in [m-1],
\]
where \(R^{m-1} x + S^{m-1} \leq 0\) are valid for \(x \in X \cap \{0,1\}^l\), and by Lemma \([B.2]\) the coefficients \(a^k, b^k\) in the inequality \(\eta \geq a^k x + b^k\) satisfy
\[
Q(x) \geq a^k x + b^k - \rho_2 \epsilon_1 \quad \forall x \in X \cap \{0,1\}^l \tag{B.23}
\]
for some constant \(\rho_2\). Therefore,
\[
\text{Opt} = \min_{x \in X \cap \{0,1\}^l} c^T x + Q(x) \geq \min_{x \in X \cap \{0,1\}^l} c^T x + \eta \quad \{ \text{s.t. } \eta \geq a^k x + b^k - \rho_2 \epsilon_1 \} \tag{B.24}
\]
\[
\geq L^m - \rho_2 \epsilon_1.
\]
Combining (B.21) and (B.24) gives
\[
U^m + \rho_1 \epsilon_1 \geq c^T x^m + \bar{Q}(x^m) \geq \text{Opt} \geq L^m - \rho_2 \epsilon_1
\]
and hence
\[
|c^T x^m + \bar{Q}(x^m) - \text{Opt}| \leq U^m - L^m + (\rho_1 + \rho_2) \epsilon_1 \leq (\rho_0 + \rho_1 + \rho_2) \epsilon_1.
\]
where last inequality is due to termination criteria of Algorithm \([7]\). This shows that the returned solution \(x^m\) satisfies the desired accuracy by setting \(K = \rho_0 + \rho_1 + \rho_2\).
It suffices to show that Algorithm 7 can terminate in a finite number of iterations. We prove it by contradiction. Suppose it does not terminate, then there exist two iteration indices \( k_1 \) and \( k_2 \) with \( k_1 < k_2 \) such that \( x^{k_1} = x^{k_2} \).

\[
L^{k_2} = c^\top x^{k_2} + \eta^{k_2} \\
\geq c^\top x^{k_1} + a^{k_1} x^{k_1} + b^{k_1} \\
= c^\top x^{k_1} + a^{k_1} x^{k_1} + b^{k_1} \\
= c^\top x^{k_1} + \widetilde{Q}(x^{k_1}) \\
= U^{k_1} = U^{k_2}, \tag{B.25}
\]

which shows that the algorithm should terminate.

**Algorithm 7** A decomposition cutting-plane algorithm for **(JMICP)**

1: Set \( m \leftarrow 1, L \leftarrow -\infty \) and \( U \leftarrow \infty \).
2: while \( U - L > \rho \bigotimes \epsilon_1 \) do
3: Solve the current first-stage MILP problem (4.1) to optimality for the main iteration \( m \) using the cutting-plane oracle from [11], and get the optimal solution \( x^m \).
4: Set \( L \leftarrow \text{obj}(4.1) \), where \( \text{obj}(4.1) \) is the optimal objective of (4.1).
5: Substitute \( x^m \) into the parametric MICP (4.2), and apply Algorithm 6 to get a \((\rho \bigotimes \epsilon_1, \epsilon_1, \lambda)\)-accurate solution of (4.2) for some constant \( \rho > 0 \). (This is achievable due to Theorem B.1.)
6: Set \( U \leftarrow c^\top x^m + \epsilon \widetilde{Q}(x^m) \), where \( \widetilde{Q}(x^m) \) is the approximated objective of (4.2) given by Algorithm 6.
7: When Algorithm 6 terminates with an approximated optimal solution to (4.2), a linear program (4.3) is established. Generate a Bender’s cut (4.4) from the dual objective of (4.3), and add it to (4.1) as \( \eta \geq a^m \top x + b^m \).
8: Set \( m \leftarrow m + 1 \).
9: end while
10: return \( x^m \) as an approximated optimal solution of **(JMICP)**.

### B.3 Analysis for the TSS-MICP problem

**Definition B.4** A feasible solution \( x \) of **(TSS-MICP)** is \( \epsilon \)-optimal if \( \text{Obj}(x) \leq \text{Opt} + \epsilon \), where \( \text{Obj}(x) \) is the objective value of **(TSS-MICP)** at \( x \) and \( \text{Opt} \) is the optimal objective of **(TSS-MICP)**.

**Theorem B.3** Let Assumption 5.9 hold. Suppose the master problem (5.5) and all scenario sub-problems (5.1) at every iteration is \((\epsilon, \lambda)\)-stable, and they are solved to \((\rho \bigotimes \epsilon, \epsilon, \lambda)\)-accuracy using Algorithm 8 for some constant \( \rho > 0 \). (This is achievable due to Theorem B.1.) Suppose the worst-case scenario detection problem (5.3) at every iteration is solved to \((\epsilon, \epsilon)\)-accuracy (Definition B.1). Then Algorithm 8 can identify a \( K \epsilon \)-optimal solution of **(TSS-MICP)** in a finite number of iterations, where \( K > 0 \) is a constant.

**Proof** The proof is similar to that of Theorem 5.9 but taking into account of optimization inaccuracy. We first show that if the algorithm terminates, it will return a \( K \epsilon \)-optimal solution of **(TSS-MICP)** for some \( K > 0 \). Suppose the algorithm terminates at iteration \( m \), and it returns a solution \( x^m \). The upper bound \( U^m \) at the termination iteration is determined by the following equation

\[
U^m = c^\top x^m + \sum_{\omega \in \Omega} \kappa^m \omega \widetilde{Q}(x^m, \xi^m), \tag{B.26}
\]
where \( \tilde{Q}(\cdot, \xi^\omega) \) is an approximated objective of \( \text{Sub}(\cdot, \xi^\omega) \) as the output of applying Algorithm 6, i.e., it is an approximated evaluation of \( Q(\cdot, \xi^\omega) \). The vector \( p^m_\omega \) is a \((\epsilon, \epsilon)\)-accurate solution of the convex program \( \max_{\omega \in \Omega} \sum_{\omega \in \Omega} p^m_\omega \tilde{Q}(x^m, \xi^\omega) \). By the assumption of the convex optimization oracle, we should have
\[
\left| \tilde{Q}(x^m, \xi^\omega) - Q(x^m, \xi^\omega) \right| \leq \rho \epsilon \quad \forall \omega \in \Omega,
\]
\[
\left| \sum_{\omega \in \Omega} p^m_\omega \tilde{Q}(x^m, \xi^\omega) - \max_{P \in \mathcal{P}} \mathbb{E}_P[\tilde{Q}(x^m, \xi)] \right| \leq \epsilon. \tag{B.27}
\]

Let \( \text{Obj}(x^m) \) be the objective value of \( \text{TSS-MICP} \) evaluated at \( x^m \). We can bound the error \( |L^m - \text{Obj}(x^m)| \) as follows:
\[
|L^m - \text{Obj}(x^m)| = \left| \sum_{\omega \in \Omega} p^m_\omega \tilde{Q}(x^m, \xi^\omega) - \max_{P \in \mathcal{P}} \mathbb{E}_P[Q(x^m, \xi)] \right|
\leq \left| \sum_{\omega \in \Omega} p^m_\omega \tilde{Q}(x^m, \xi^\omega) - \max_{P \in \mathcal{P}} \mathbb{E}_P[\tilde{Q}(x^m, \xi)] \right| + \left| \max_{P \in \mathcal{P}} \mathbb{E}_P[\tilde{Q}(x^m, \xi)] - \max_{P \in \mathcal{P}} \mathbb{E}_P[Q(x^m, \xi)] \right|
\leq \epsilon + \rho \epsilon. \tag{B.28}
\]

Let \( \text{Opt}_m \) be the exact optimal objective value of (B.34) at iteration \( m \). Since \((x^m, \eta^m)\) is a \((\rho \epsilon, \epsilon, \lambda)\)-accurate solution of (B.34), we have \( \text{Opt}_m \geq L - \rho \epsilon \), where \( L = c^\top x^m + \eta^m \). Furthermore, the strong duality implies that
\[
\tilde{Q}(x^k, \xi^\omega) = p^k_\omega x^k + u^k \quad \forall \omega, k.
\]

Note that since each scenario sub-problem is a JMIPC, we can apply Lemma B.2 to get the following set of inequalities for some constant \( \rho_1 > 0 \):
\[
r^k_\omega x + u^k \leq Q(x, \xi^\omega) + \rho_1 \epsilon \quad \forall \omega, k, x,
\]
\[
r^k_\omega x + u^k \geq Q(x, \xi^\omega) - \rho_1 \epsilon \quad \forall \omega, k. \tag{B.29}
\]

The inequality (B.28) further indicates that
\[
\max_{P \in \mathcal{P}} \mathbb{E}_P[Q(x^k, \xi)] - (\rho + 1) \epsilon \leq \sum_{\omega \in \Omega} p^k_\omega Q(x^k, \omega) \leq \max_{P \in \mathcal{P}} \mathbb{E}_P[Q(x^k, \xi)] + (\rho + 1) \epsilon. \tag{B.30}
\]

Claim: Let \( x^* \) be the optimal solution of \( \text{TSS-MICP} \) and \( \eta^* = \max_{P \in \mathcal{P}} \mathbb{E}_P[Q(x^*, \xi)] + \rho_1 \epsilon \). Then \((x^*, \eta^*)\) is a feasible solution of (B.34) at any master iteration.

Proof of the claim: it suffices to verify that the following inequality holds for each \( k \)
\[
\max_{P \in \mathcal{P}} \mathbb{E}_P[Q(x^*, \xi)] + \rho_1 \epsilon \geq \sum_{\omega \in \Omega} p^k_\omega (x^* + u^k) - \rho_1 \epsilon. \tag{B.31}
\]

Indeed, we have
\[
\max_{P \in \mathcal{P}} \mathbb{E}_P[Q(x^*, \xi)] + \rho_1 \epsilon \geq \sum_{\omega \in \Omega} p^k_\omega (x^* + u^k) + \rho_1 \epsilon \quad \text{\( \geq \)} \quad \sum_{\omega \in \Omega} p^k_\omega (x^* + u^k) - \rho_1 \epsilon \quad \text{\( \geq \)} \quad \sum_{\omega \in \Omega} p^k_\omega (x^* + u^k), \tag{B.29}
\]

which concludes the proof of the claim.

Let \( \text{Opt} \) be the optimal objective of \( \text{TSS-MICP} \). The claim indicates that
\[
\text{Opt} = c^\top x^* + \max_{P \in \mathcal{P}} \mathbb{E}_P[Q(x^*, \xi)]
= c^\top x^* + \eta^* - \rho_1 \epsilon
\geq \text{Opt}_m - \rho_1 \epsilon, \tag{B.32}
\]
where $L^m$ is the value of $L$ at the iteration $m$. Therefore, the following inequalities hold

$$\text{Obj}(x^m) \leq U^m + (1 + \rho)\epsilon \quad \text{(using (B.28))}$$
$$\leq L^m + \rho \epsilon + (1 + \rho)\epsilon \quad \text{(termination criteria)}$$
$$\leq L^m + \rho \epsilon + \rho \epsilon + (1 + \rho)\epsilon \quad \text{(using (B.32))}$$

The above shows that $x^m$ is a $(1 + \rho + \rho + \rho)\epsilon$-optimal solution.

Next we show that the algorithm terminates in a finite number of iterations. Suppose it does not terminate, then there exist two iteration indices $k_1$ and $k_2$ with $k_1 < k_2$ such that $x^{k_1} = x^{k_2}$. Then

$$L^{k_2} = c^T x^{k_2} + \eta^{k_2} \geq c^T x^{k_2} + \sum_{\omega \in \Omega} \rho^{k_1 \omega} x^{k_2} + \sum_{\omega \in \Omega} \rho^{k_1 \omega} \epsilon^{k_2 \omega} \geq c^T x^{k_1} + \sum_{\omega \in \Omega} \rho^{k_1 \omega} \tilde{Q}(x^{k_1}, \xi^{k_2}) \geq U^{k_1} = U^{k_2}$$

which shows that the algorithm should terminate.

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. Cornuëjols, G.: Valid inequalities for mixed integer linear programs. Mathematical Programming 112(1), 3–44 (2008)
4. Balas, E., Ceria, S., Cornuëjols, G., Natraj, N.: Gomory cuts revisited. Operations Research Letters 19(1), 1–9 (1996)
5. 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)
6. Padberg, M.W., Roy, T.J.V., Wolsey, L.A.: Valid linear inequalities for fixed charge problems. Operations Research 33(4), 842–861 (1985)
7. Roy, T.J.V., Wolsey, L.A.: Valid inequalities and separation for uncapacitated fixed charge networks. Operations Research Letters 4(3), 105–112 (1985)
8. Balas, E., Ceria, S., Cornuëjols, G.: A lift-and-project cutting plane algorithm for mixed 0-1 programs. Mathematical Programming 58, 295–324 (1993)
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. Chen, B., Süge Küçükyavuz, S.S.: Finite disjunctive programming characterizations for general mixed-integer linear programs. Operations Research 59(1), 202–210 (2011)
12. 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](https://arxiv.org/pdf/0707.3945.pdf)
13. Duran, M., Grossmann, I.: An outer-approximation algorithm for a class of mixed integer nonlinear programs. Mathematical Programming 36(3), 307–339 (1986)
14. Lubin, M., Yamangil, E., Bent, R., Vielma, J.P.: Polyhedral approximation in mixed-integer convex optimization. Mathematical Programming pp. 139–168 (2018)
15. Gade, D., Küçükyavuz, S., Sen, S.: Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. Mathematical Programming 144, 39–64 (2014)
Algorithm 8: A decomposition cutting-plane algorithm for solving (TSS-MICP) to a given accuracy.

1: Initialization: $L \leftarrow -\infty$, $U \leftarrow \infty$, $m \leftarrow 1$.
2: while $U - L > \rho \varepsilon$ do
3: Solve the following approximated first-stage master problem to obtain $(\eta^m, x^m)$ using Algorithm 6
4: for some constant $\rho$; (This is achievable due to Theorem B.1.)
5: \[
\begin{aligned}
\min_x \quad & c^T x + \eta \\
\text{s.t.} \quad & Ax \leq b, \\
& \eta \geq \sum_{\omega \in \Omega} p_{k\omega}^x k_{\omega} x + \sum_{\omega \in \Omega} p_{k\omega} u_{k\omega} \quad \forall k \in [m - 1], \\
& x \in C \cap \{0, 1\}^n,
\end{aligned}
\] (B.34)
6: where $\{p_{k\omega}^x : \omega \in \Omega\}$ is a $(\varepsilon, \varepsilon)$-accurate solution of (5.3) obtained by the convex optimization oracle. (See Line 12.) The coefficients $r_{k\omega}$ and $u_{k\omega}$ are specified in Line 9.
7: Update the lower bound as $L \leftarrow c^T x^m + \eta^m$.
8: for $\omega \in \Omega$ do
9: Solve the second-stage problem Sub($x^m$, $\omega$) to get a $(\rho \varepsilon, \varepsilon, \lambda)$-accurate solution using Algorithm 6 for some constant $\rho$; and let $Q(x^m, \xi^\omega)$ be the approximated objective given by the algorithm. (This is achievable due to Theorem B.1.) When Algorithm 6 terminates, it generates a LP relaxation in the form of (5.2) with $\hat{x} = x^m$, which yields an approximated Bender’s inequality written as $Q(x^m, \xi^\omega) + \rho \varepsilon \geq r_{m\omega}^x \hat{x} + u_{m\omega}$.
10: end for
11: Solve (5.3) with $\hat{x} = x^m$ (using the convex-optimization oracle) to get a $(\varepsilon, \varepsilon)$-accurate solution but satisfying $\sum_{\omega \in \Omega} p_{k\omega}^m = 1$. (Note that the normalization equality can be achieved by perturbing entries of the $p_{k\omega}^m$ vector.)
12: which is denoted as $p_{k\omega}^m = \{p_{k\omega}^m : \omega \in \Omega\}$.
13: Generate the following aggregated Benders cut with iteration index $m$:
14: \[
\begin{aligned}
\eta \geq \sum_{\omega \in \Omega} p_{k\omega}^m k_{\omega} x + \sum_{\omega \in \Omega} p_{k\omega}^m u_{k\omega},
\end{aligned}
\] and add it to (B.34).
15: Update the upper bound as $U \leftarrow c^T x^m + \sum_{\omega \in \Omega} p_{k\omega}^m Q(x^m, \xi^\omega)$.
16: $m \leftarrow m + 1$.
17: end while
18: Return $x^m$.

16. Zhang, M., Küçükyavuz, S.: Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs. SIAM Journal on Optimization 24(4), 1933–1951 (2014)
17. 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)
18. 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/s10898-020-00966-w
19. Atamtürk, A., Narayanan, V.: Conic mixed-integer rounding cuts. Mathematical Programming 122, 1–20 (2008)
20. Sen, S., Sherali, H.D.: Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. Mathematical Programming 106, 203–223 (2006)
21. Luo, F., Mehrotra, S.: A decomposition method for distributionally-robust two-stage stochastic mixed-integer conic programs. Mathematical Programming (2021). URL https://doi.org/10.1007/s10107-021-01641-2
22. Rockafellar, R.T.: Convex Analysis. Princeton University Press (1970)
23. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization - I. Springer-Verlag (1993)