Semi-Infinite Mixed Binary and Disjunctive Programs: Applications to Set-Covering with Infinite Demand Points and Implicit Hitting Set Problems[1]

Manish Bansal
Grado Department of Industrial and Systems Engineering
Virginia Tech, Blacksburg, VA 24060 {bansal@vt.edu}

Abstract. Sherali and Adams [Discrete Applied Math. 157: 1319-1333, 2009] derived convex hull of semi-infinite mixed binary linear programs (SIMBLPs) using Reformulation-Linearization Technique (RLT). In this paper, we study semi-infinite disjunctive programs (SIDPs – a generalization of SIMBLPs) and present linear programming equivalent and valid inequalities for them. We utilize these results for deriving a hierarchy of relaxations for SIMBLPs along with solution approaches for them. This also establishes a direct connection between RLT and linear programming equivalent for disjunctive programs, even without sequential convexification and the requirement of computing projections multiple times. Additionally, we present an exact algorithm for SIBLPs with implicity defined constraints (SIBLP-IC), and formulate set-covering problem with infinite number of demand points or spatial representation of demand as SIBLP-IC. Based on our computational results for solving the set-covering (and equivalent implicit hitting set) problem instances, we observe that the foregoing approach is computationally efficient in comparison to Gurobi 9.5.2 and an algorithm of Moreno-Centeno and Karp [Operations Research 61(2): 453-468, 2013] for implicit hitting set problem (a special case of SIBLP-IC).

Keywords. Semi-infinite mixed binary program; Disjunctive program; RLT; Hierarchy of relaxations; Set covering problem; Infinite demand points; Implicit hitting-set problem.

1 Introduction and Motivation

Theory and algorithms for semi-infinite linear programs, i.e., linear programs with infinite number of constraints but finite number of variables, have been well established [Charnes et al., 1963, 1965, Goberna and Lopez-Cerda, 1998]. These programs arise in numerous applications such as robotic control, finance, mechanics of materials, eigenvalue computations, and many more [Hettich and Kortanek, 1993]. A natural extension of semi-infinite linear programs is the incorporation of discrete variables. Sherali and Adams [2009] studied semi-infinite mixed binary linear programs (SIMBLPs) and utilized Reformulation-Linearization Technique (RLT) to derive its equivalent semi-infinite linear program. In this paper, we study semi-infinite disjunctive linear programs (SIDPs) that subsume SIMBLPs. We present linear programming equivalent and valid inequalities for SIDPs defined by disjunctions/union of semi-infinite linear programs in Section 2. Using these results for SIDPs, we derive a hierarchy of relaxations for SIMBLPs, showcase that RLT based reformulation

[1]First Draft: 23 Nov. 2022

can be directly derived using SIDPs, and present solution approaches for SIMBLPs (Section 3). As per our knowledge, this direct connection between RLT and disjunctive programs is not known in the literature, even for finite cases. Note that disjunctive convex programs, SIBLPs with discrete variables, robust 0-1 knapsack problems, distributionally robust programs, and distributionally robust chance-constraint programs can be formulated as SIDP as well (refer to Section 3.4).

We also study semi-infinite binary linear programs with implicitly defined constraints (SIBLP-IC), i.e., all constraints are not explicity known but given a point, there exists an oracle to either certify that all constraints are satisfied or return a violated constraint. We present a branch-and-cut approach to solve SIBLP-IC exactly (Section 4). Various combinatorial optimization problems can be reformulated as (SI)BLP-IC. Examples include (but not limited to), set covering problem with infinite number of demand points, maximum feasible subsystem [Chinneck, 2001], facility location, multi-genome alignment [Moreno-Centeno and Karp, 2013], machine learning [Chinneck, 2019], matriod intersection, and vehicle routing. As a consequence, the results for the studied semi-infinite programs have a wide range of applicability. In this paper, as a special case of SIBLP, we consider the set-covering problem with infinite number of demand points or spatial representation of demand points.

Set Covering Problem with Infinite Demand Points and Logical Constraints. (Explicit) set covering problem is a well-known combinatorial optimization problem that is defined as follows. Given a finite ground set U along with a set of subsets of U, denoted by $\mathcal{T}$, the objective of a decision maker is to find a minimum number of elements of $\mathcal{T}$ such that the union of these subsets is the ground set. Each element of the ground set represents a demand point, and an element of $\mathcal{T} := \{T_1, \ldots, T_m\}$ denotes the set of demand points that can be served by a potential facility location. This problem can be formulated as a binary program with variables $\eta_i = 1$, if subset or facility location T_i is selected, and 0 otherwise, for $i \in \{1, \ldots, m\}$. A major concern with this framework is the binary coverage assumption, i.e., a region represented by a demand point (its centroid) is assumed to be either completely covered or not covered at all. This leads to misrepresentation errors because it does not allow partial coverage of the (spatial) region [Murray, 2005]. In other words, coverage (or non-coverage) of a representative demand point by a facility does not imply that the entire region is covered (or not covered) by the facility. One way to address this issue is to consider spatial representation of the demand such as rectangle, circle, polytope, etc., thereby leading to a challenging class of computational geometry problems [Bansal and Kianfar, 2017].

We propose to tackle this challenge by considering infinite number of demand points to represent a spatial region and formulate associated set-covering problem as SIBLP-IC. Specifically, we consider a ground set U with infinite number of points, and the subsets $T_i \subseteq U$, $i = 1, \ldots, m$, can be of finite/infinite size with weight w_i. We also allow inclusion of a finite number of logical linear constraints that are user/problem specific, denoted by $A_L\eta \geq b_L$ (refer to Remark 1 for an example). The goal is still to find set $S \subseteq \{1, \ldots, m\}$ of minimum total weight ($\sum_{i \in S} w_i$) such that

$\cup_{i \in S} T_i = U$. This problem can also be formulated as the following SIBLP:

$$\min_{\eta \in \{0,1\}^m} \left\{ \sum_{i=1}^{m} w_i \eta_i : A_L \eta \geq b_L, \quad \sum_{\substack{i=1 \\ u \in T_i}}^{m} \eta_i \geq 1, \quad \text{for } u \in U \right\}. \tag{1}$$

Here, for each $u \in U$, there is a linear constraint to ensure that at least one facility location $i \in \{1, \ldots, m\}$ covering u, i.e., $u \in T_i$, is selected. The aforementioned approaches for SIBLP-IC and SIMBLP solve this problem exactly along with providing partial convex hulls and valid inequalities for its feasible region.

In the context of implicity defined combinatorial optimization problems, Moreno-Centeno and Karp [2013] introduced an exact algorithm for Implicit Hitting Set Problem. Specifically, (explicit) hitting set problem is another well-known combinatorial optimization problems where given a finite ground set $G := \{1, \ldots, m\}$ and a set of subsets of G (denoted by $\mathcal{S}$), the goal is to find a hitting set $H \subseteq G$ of minimum cardinality such that H "hits" or contains at least one element as a representative from each subset belonging to $\mathcal{S}$. This is an NP-complete problem [Karp, 1972] and it can be formulated as the following binary linear program with a finite number of binary variables (m) and linear constraints ($|\mathcal{S}|$):

$$\min_{\mathbf{x} \in \{0,1\}^m} \left\{ \sum_{i \in G} x_i : \sum_{i \in C} x_i \geq 1 \text{ for all } C \in \mathcal{S} \right\}, \tag{2}$$

where $x_i = 1$ when $i \in H$, and 0 otherwise. Notice that it is a special case of (1). In implicit hitting set problem (IHSP), the set $\mathcal{S}$ is not explicitly defined, and it is assumed that there exists an oracle to verify that all subsets in $\mathcal{S}$ are hit by a given set H or otherwise, the oracle returns a subset $C \in \mathcal{S}$ not hit by H. This results in a binary program with implicitly defined constraints, which is a special case of SIBLP-IC. In Section 4.2, we evaluate the performance of our algorithm for SIBLP-IC to solve this problem, which is equivalent to the set-covering problem (1) with implicity defined demand points, unit weights w_i, and no additional logical constraints.

Remark 1 (Covering Tour Problem – A Special Case of SIBLP-IC) *Gendreau et al. [1997] amalgamated explicit set-covering and traveling salesperson problems to get Covering Tour Problem, also referred to as Traveling Circus Problem in the literature. This problem is defined for a finite set of vertices to be covered by the salesperson on tour. A vertex is assumed to be covered if it is within some distance from any of the toured vertices. The authors mentioned its applications in locating "post (or ballot) boxes among a set of candidate sites in such a way that all users are located within reasonable distance of a post box and that the cost of a collection route through all post boxes is minimized." For the problem with a large number of users, it can be formulated as* (1) *along with other constraints of traveling salesperson problem; refer to formulation (1)-(7) in Gendreau et al. [1997] for details.*

2 Semi-Infinite Disjunctive Programs

In this section, we present linear programming equivalent and valid inequalities for SIDPs defined by disjunctions of semi-infinite linear programs:

$$\min_{\mathbf{x}\in\mathbb{R}^n_+} \left\{ \sum_{i=1}^{n} c_i x_i : \bigvee_{h\in H} \left(\sum_{i=1}^{n} a_{ij}^h x_i \geq b_j^h \text{ for all } j \in J \right) \right\}, \tag{SIDP}$$

where $\vee$ denotes disjunction/union/OR logic operator, and $|H|$ is finite but $|J|$ is not finite. Balas [1979, 1998] provided a tight extended formulation for convex hull of (SIDP) with finite J. In this section, we first present its extension for (SIDP). Let

$$\mathcal{D} := \left\{ \mathbf{x} \in \mathbb{R}^n : \bigvee_{h\in H} \mathcal{D}_h \right\} \text{ where } \mathcal{D}_h := \left\{ \mathbf{x} \in \mathbb{R}^n_+ : \sum_{i=1}^{n} a_{ij}^h x_i \geq b_j^h \text{ for all } j \in J \right\}$$

for $h \in H$, is a nonempty and bounded semi-infinite linear program, and a_{ij}^h, b_j^h are rational numbers.

Theorem 1 *The convex hull of $\mathcal{D}$, denoted by $conv(\mathcal{D})$, is equivalent to the projection of the following extended formulation (denoted by $\mathcal{R}$) onto the x-space:*

$$\mathbf{x} = \sum_{h\in H} \zeta^h, \tag{3a}$$

$$\sum_{i=1}^{n} a_{ij}^h \zeta_i^h \geq b_j^h \zeta_0^h, \quad h \in H \text{ and } j \in J, \tag{3b}$$

$$\sum_{h\in H} \zeta_0^h = 1, \tag{3c}$$

$$(\zeta^h, \zeta_0^h) \geq 0, \quad h \in H. \tag{3d}$$

Proof. First, we prove that the projection of $\mathcal{R}$ onto the x-space, $\text{Proj}_x(\mathcal{R}) \subseteq conv(\mathcal{D})$. Let $\left(\hat{\mathbf{x}}, \{\hat{\zeta}^h, \hat{\zeta}_0^h\}_{h\in H}\right) \in \mathcal{R}$. Observe that in case $\hat{\zeta}_0^h = 0$ for any $h \in H$, $\sum_{i=1}^{n} a_{ij}^h \hat{\zeta}_i^h \geq 0$, $j \in J$. However, because of boundedness of $\mathcal{D}_h$, there does not exist any direction/vector such that

$$\sum_{i=1}^{n} a_{ij}^h d_i^h \geq 0 \text{ for all } j \in J, \quad \mathbf{d}^h \geq \mathbf{0}, \text{ and } \mathbf{d}^h \neq \mathbf{0}.$$

This implies that $\hat{\zeta}^h = 0$ whenever $\hat{\zeta}_0^h = 0$. Therefore, without any loss of generality, we assume that $\hat{\zeta}_0^h > 0$ for all $h \in H$. From inequalities (3b) and (3d), it is evident that $\hat{\zeta}^h/\hat{\zeta}_0^h \in \mathcal{D}_h$ which is a nonempty convex set. For simplicity, let $\hat{\zeta}^h/\hat{\zeta}_0^h = \hat{\eta}^h \in \mathcal{D}_h$. Now, from inequalities (3a) and (3c),

we get

$$\hat{\mathbf{x}} = \sum_{h \in H} \hat{\zeta}_0^h \hat{\eta}^h, \quad \sum_{h \in H} \hat{\zeta}_0^h = 1, \quad \hat{\zeta}_0^h \geq 0.$$

This implies that $\hat{\mathbf{x}}$ can be written as convex combination of points $\{\hat{\eta}^h\}_{h \in H}$ where $\hat{\eta}^h \in \mathcal{D}_h$. Hence, $\hat{\mathbf{x}} \in conv(\cup_h \mathcal{D}_h) = conv(\mathcal{D})$.

We prove $conv(\mathcal{D}) \subseteq \text{Proj}_x(\mathcal{R})$ by showcasing that for each point $\tilde{\mathbf{x}} \in conv(\mathcal{D})$, there exists a vector $\left(\tilde{\mathbf{x}}, \{\tilde{\zeta}^h, \tilde{\zeta}_0^h\}_{h \in H}\right) \in \mathcal{R}$. This part of the proof is rather straightforward. Let $\tilde{\mathbf{x}} = \sum_{h \in H} \lambda_h \mathbf{x}^h$ such that $\sum_{h \in H} \lambda_h = 1$, $\mathbf{x}^h \in \mathcal{D}_h$, and $\lambda_h \geq 0$ for all $h \in H$. We set $\tilde{\zeta}^h = \lambda_h \mathbf{x}^h$ and $\tilde{\zeta}_0^h = \lambda_h$ to get $\left(\tilde{\mathbf{x}}, \{\tilde{\zeta}^h, \tilde{\zeta}_0^h\}_{h \in H}\right) \in \mathcal{R}$. This completes the proof. □

Remark 2 *Notice that $\mathcal{D}_h$ is a convex set and $\mathcal{D}$ is defined by disjunction of convex sets. Therefore, Theorem 1 can also be proved using Proposition 3.3.5 in Ben-Tal and Nemirovski [2001] for union of general convex sets. However, this result is not known in the literature for SIDPs specifically. Another motivation behind stating this theorem (with simpler proof) are two folds. First, it leads to the derivation of Theorem 1 of Sherali and Adams [1990] and Theorem 1 of Sherali and Adams [2009], thereby establishing direct connection between RLT and disjunctive programming. Second, we use it to derive hierarchy of relaxations and other solution approaches for SIMBLPs in Section 3.*

Next, we present projection of a semi-infinite linear program onto a lower-dimensional space that can be utilized to get projection of the aforementioned extended formulation of SIDPs. Additionally, we provide a class of valid inequalities for SIDPs. These results extend similar known results for finite linear and disjunctive programs.

Definition 1 *Let $Q = \{(x, \eta) \in \mathbb{R}^n \times \mathbb{R}^p : \sum_{i=1}^n a_{ij}^1 x_i + \sum_{i=1}^p a_{ij}^2 \eta_i \leq b_j, j \in J\}$. Then, projection of Q on to x-space is defined as:*

$$Proj_x(Q) = \{x \in \mathbb{R}^n : \textit{there exists } \eta \in \mathbb{R}^p \textit{ such that } (x, \eta) \in Q\}. \tag{4}$$

Proposition 1 *For $\widehat{J} \subseteq J$, let*

$$W(\widehat{J}) := \left\{ v \in \mathbb{R}_+^{|\widehat{J}|} : \sum_{j \in \widehat{J}} v_j a_{ij}^2 = 0, \quad i = 1, \ldots, n \right\}.$$

Then, $Proj_x(Q) = \mathcal{R}$ where

$$\mathcal{R} = \left\{ x \in \mathbb{R}^n : \sum_{i=1}^n \left(\sum_{j \in \widehat{J}} v_j a_{ij}^1 \right) x_i \leq \sum_{j \in \widehat{J}} v_j b_j \textit{ for all } v \in W(\widehat{J}) \textit{ and } \widehat{J} \subseteq J \right\}.$$

Proof. Given $\hat{x} \in \mathcal{R}$, assume that there does not exist $\hat{\eta} \in \mathbb{R}^p$ such that $(\hat{x}, \hat{\eta}) \in Q$. This implies

there exists index subset $\widehat{J} \subseteq J$ such that

$$\sum_{i=1}^{n} a_{ij}^1 \hat{x}_i + \sum_{i=1}^{p} a_{ij}^2 \hat{\eta}_i > b_j \text{ for } j \in \widehat{J}.$$

Multiplying these inequalities with a vector $v \in W(\widehat{J})$ from left side results in

$$\sum_{i=1}^{n} \sum_{j \in \widehat{J}} v_j a_{ij}^1 \hat{x}_i > \sum_{j \in \widehat{J}} v_j b_j,$$

which contradicts $\hat{x} \in \mathcal{R}$. Hence, $\hat{x} \in \text{Proj}_x(Q)$ and $\mathcal{R} \subseteq \text{Proj}_x(Q)$. On the other hand, let $(\hat{x}, \hat{\eta}) \in Q$. Now, multiplying each defining inequality with $v \in W(\widehat{J})$ for $\widehat{J} \subseteq J$ leads to $\hat{x} \in \mathcal{R}$. □

Proposition 2 *Assume that $\mathcal{D}^h$ is nonempty and the set of points $\{(a_j^h, b_j^h) \in \mathbb{R}^{n+1}\}_{j \in J}$ is closed and bounded, for each $h \in H$. An inequality $\pi^T x \geq \pi_0$ is valid for $\mathcal{D}$ if and only if there exists vector $\{\sigma^h \in \mathbb{R}_+^m\}_{h \in H}$ such that*

$$\pi_i \geq \sum_{j \in J_m^h} \sigma_j^h a_{ij}^h \quad \textit{for } i = 1, \ldots, n, \textit{ and } \pi_0 \leq \sum_{j \in J_m^h} \sigma_j^h b_j \quad \textit{for all } h \in H, \tag{5}$$

where $J_m^h \subseteq J$ and $|J_m^h| = m \leq n$.

Proof. According to Theorem 17.3 of Rockafellar [1970], inequality $\pi^T x \geq \pi_0$ is valid for $\mathcal{D}^h$ (a nonempty convex set) if and only if $\pi_i = \sum_{j \in J_m^h} \sigma_j^h a_{ij}^h$ for $i = 1, \ldots, n$, and $\pi_0 \leq \sum_{j \in J_m^h} \sigma_j^h b_j$ for some finite index set $J_m^h \subset J$. As a consequence, $\pi^T x \geq \pi_0$ is valid for $\mathcal{D}$ (or $\mathcal{D}^h$ for all $h \in H$) if and only if inequalities (5) hold. □

2.1 Illustrative Numerical Example of SIDP

Consider the following (simple) two-dimensional semi-infinite linear programs for illustration:

$$\begin{aligned}
\mathcal{D}_1 &= \big\{\mathbf{x} \in \mathbb{R}_+^2 : x_1 \sin t + x_2 \cos t \leq \beta + \alpha_1 \sin t + \alpha_2 \cos t, \quad t \in [0, 2\pi]\big\}, \\
\mathcal{D}_2 &= \big\{\mathbf{x} \in \mathbb{R}_+^2 : x_2 \geq -1; \quad 2tx_1 + x_2 \leq t^2, \quad t \in [-1, 1]\big\}, \\
\mathcal{D}_3 &= \big\{\mathbf{x} \in \mathbb{R}_+^2 : \eta_1 x_1 \sin t + \eta_2 x_2 \cos t \leq 1, \quad t \in [0, 2\pi]\big\},
\end{aligned}$$

where α_1, α_2, β, η_1, and η_2 are given. (Set $\mathcal{D}_2$ is Example 5.3 in a survey paper by Hettich and Kortanek [1993].) Using Theorem 1, we derive tight extended formulations for convex hull of the following disjunctive sets.

(i) $\mathcal{D}_{123} = \mathcal{D}_1 \vee \mathcal{D}_2 \vee \mathcal{D}_3$:

$$\begin{aligned}
&\zeta_1^1 \sin t + \zeta_2^1 \cos t \le (\beta + \alpha_1 \sin t + \alpha_2 \cos t)\, \zeta_0^1, \quad t \in [0, 2\pi], \qquad \zeta_2^2 \ge -\zeta_0^2; \\
&2t\zeta_1^2 + \zeta_2^2 \le t^2 \zeta_0^2, \quad t \in [-1, 1]; \qquad \eta_1 \zeta_1^3 \sin t + \eta_2 \zeta_2^3 \cos t \le \zeta_0^3, \quad t \in [0, 2\pi] \\
&\zeta_0^1 + \zeta_0^2 + \zeta_0^3 = 1; \qquad \mathbf{x} = \zeta^1 + \zeta^2 + \zeta^3; \qquad (\zeta^h, \zeta_0^h) \ge 0, h = 1, 2, 3.
\end{aligned}$$

(ii) $\mathcal{D}_{23} := \mathcal{D}_2 \vee \mathcal{D}_3$:

$$\begin{aligned}
&2t\zeta_1^2 + \zeta_2^2 \le t^2 \zeta_0^2, \quad t \in [-1, 1]; \qquad \eta_1 \zeta_1^3 \sin t + \eta_2 \zeta_2^3 \cos t \le \zeta_0^3, \quad t \in [0, 2\pi] \\
&\zeta_2^2 \ge -\zeta_0^2; \quad \zeta_0^2 + \zeta_0^3 = 1; \qquad \mathbf{x} = \zeta^2 + \zeta^3; \qquad (\zeta^h, \zeta_0^h) \ge 0, \ h = 2, 3.
\end{aligned}$$

(iii) $\mathcal{D}_{13} := \mathcal{D}_1 \vee \mathcal{D}_3$:

$$\begin{aligned}
&\zeta_1^1 \sin t + \zeta_2^1 \cos t \le (\beta + \alpha_1 \sin t + \alpha_2 \cos t)\, \zeta_0^1, \ t \in [0, 2\pi]; \quad \mathbf{x} = \zeta^1 + \zeta^3; \\
&\eta_1 \zeta_1^3 \sin t + \eta_2 \zeta_2^3 \cos t \le \zeta_0^3, t \in [0, 2\pi] \quad \zeta_0^1 + \zeta_0^3 = 1; \quad (\zeta^h, \zeta_0^h) \ge 0, h = 1, 3.
\end{aligned}$$

Figure 1 provides geometrical representation of $\mathcal{D}_{23}$ and three special cases of $\mathcal{D}_{13}$ in (x_1, x_2) space: (a) $\beta \ge \max\{\eta_1^{-1}, \eta_2^{-1}\}$ and $\alpha_1 = \alpha_2 = 0$, (b) $\beta \le \max\{\eta_1^{-1}, \eta_2^{-1}\}$ and $\alpha_1 = \alpha_2 = 0$, and (c) $\eta_1 = \eta_2 = \beta^{-1}$, $\alpha_1 \ge \eta_1^{-1} + \beta$, and $\alpha_2 = 0$. Their convex hulls are presented by colored regions.

Even algebraically, it is easy to observe that $\mathcal{D}_{13}$ reduces to $\mathcal{D}_1$ and $\mathcal{D}_3$ for special cases (a) and (b), respectively. For case (c), we project the following tight extended formulation of $\mathcal{D}_{13}$ onto (x_1, x_2) space:

$$\zeta_1^1 \sin t + \zeta_2^1 \cos t \le (\beta + \alpha_1 \sin t)\, \zeta_0^1, \ t \in [0, 2\pi]; \tag{6}$$

$$\zeta_1^3 \sin t + \zeta_2^3 \cos t \le \beta \zeta_0^3, \quad t \in [0, 2\pi]; \tag{7}$$

$$\mathbf{x} = \zeta^1 + \zeta^3; \quad \zeta_0^1 + \zeta_0^3 = 1; \quad (\zeta^h, \zeta_0^h) \ge 0, h = 1, 3. \tag{8}$$

Specifically, for $t \in [\pi, 2\pi]$, we aggregate (6) and (7) to get

$$x_1 \sin t + x_2 \cos t \le \beta + (\alpha_1 \sin t)\, \zeta_0^1 \le \beta. \tag{9}$$

By setting $t = \pi$ and $t = 2\pi$ in (9), we get $x_2 \ge -\beta$ and $x_2 \le \beta$, respectively. Likewise, for $t \in [0, \pi]$, aggregating (6) and (7) gives

$$x_1 \sin t + x_2 \cos t \le \beta + (\alpha_1 \sin t)\, \zeta_0^1 \le \beta + \alpha_1 \sin t(\zeta_0^1 + \zeta_0^3) \le \beta + \alpha_1 \sin t. \tag{10}$$

Hence, $conv(\mathcal{D}_{13}) = \{(x_1, x_2) : (9), (10), |x_2| \le \beta\}$ for case (c).

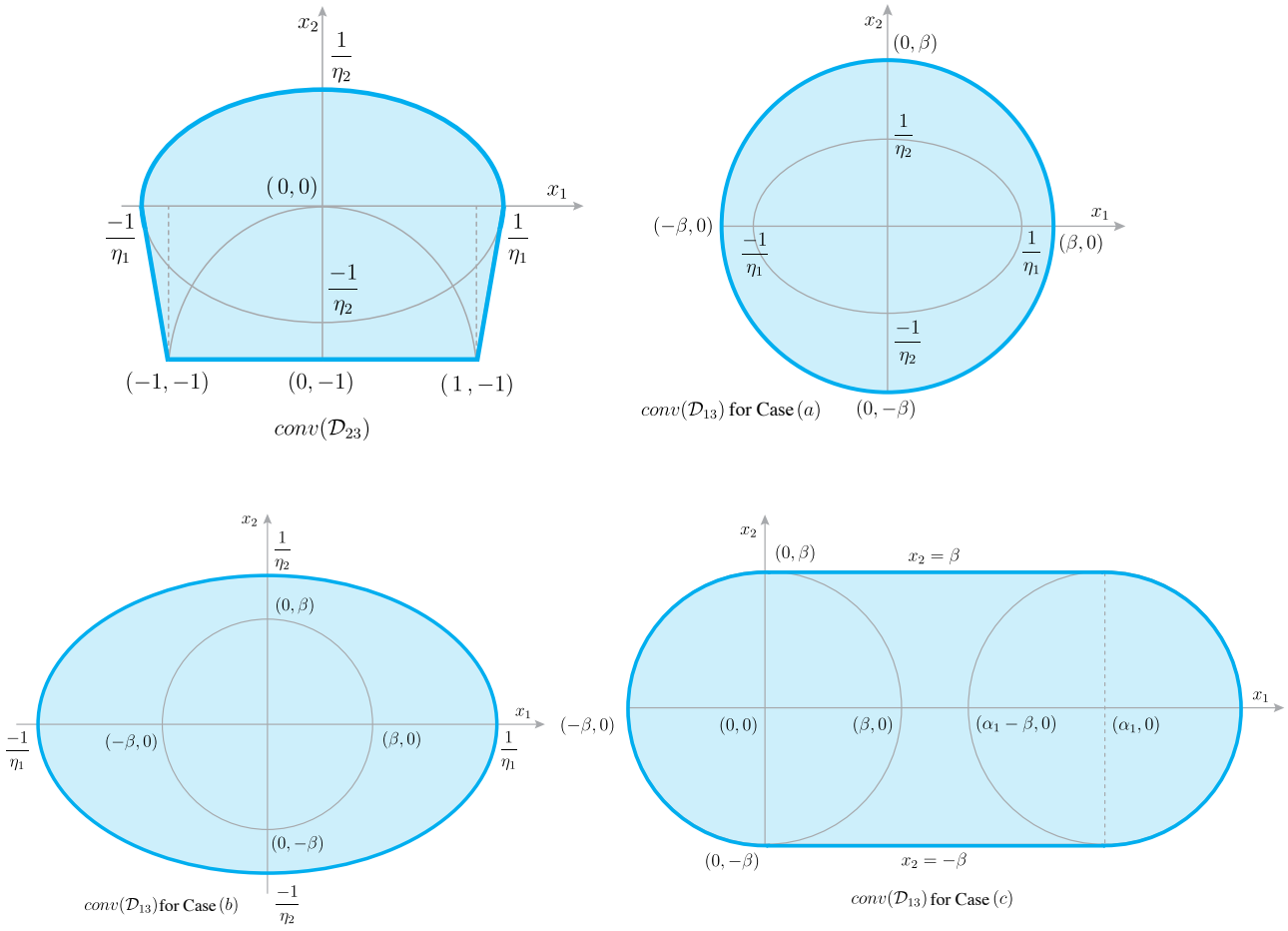

Figure 1: Geometrical representation of $\mathcal{D}_{23}$ and three special cases of $\mathcal{D}_{13}$ in (x_1, x_2) space

3 Semi-Infinite Mixed Binary Linear Programs

The SIMBLPs are defined as follows:

$$\min_{\mathbf{x}\in[0,1]^n} \left\{ \sum_{i=1}^{n} c_i x_i : \left(\sum_{i=1}^{n} a_{ij} x_i \geq b_j, j \in J \right) \wedge (x_i = 0 \vee x_i = 1 \text{ for } i = 1, \ldots, p) \right\}, \tag{11}$$

where the defining constraints include $0 \leq x_i \leq 1$ for $i = 1, \ldots, n$ (for simplicity and wlog). We denote the feasible region of (11) by $\mathcal{X}^M$ and its LP relaxation by $\mathcal{X}^M_{LP}$.

3.1 Derivation of RLT-based reformulation of Sherali and Adams [2009] using Theorem 1 and Hierarchy of Relaxations for SIMBLPs

Sherali and Adams [2009] introduced convex hull representation of $\mathcal{X}^M$ using RLT. They multiply each defining constraint with $\left(\prod_{k\in K_1} x_k\right)\left(\prod_{k\in K_2}(1-x_k)\right)$ for each pair of disjoint sets (K_1, K_2) such that $|K_1 \cup K_2| = p$, thereby getting a monolithic polynomial reformulation. They linearize this

reformulation, and then project it to get convex hull of $\mathcal{X}^M$. We showcase how result (Theorem 1) of Sherali and Adams [2009] can be derived using Theorem 1 of this paper, even without the requirement of computing projections multiple times. First, we demonstrate for $p = 2$ and then extend it for any $p \geq 1$.

For $p = 2$, the disjunctive constraints ($x_i = 0 \vee x_i = 1$ for $i = 1, \ldots, 2$) of $\mathcal{X}^M$ are equivalent to

$$(x_1 = 0, x_2 = 0) \vee (x_1 = 1, x_2 = 0) \vee (x_1 = 0, x_2 = 1) \vee (x_1 = 1, x_2 = 1)\,.$$

As a result, set $\mathcal{X}^M$ with $p = 2$ in "disjunctive normal form" is written as:

$$\left\{ \mathbf{x} \in \mathbb{R}^n_+ : \bigvee_{(\alpha_1,\alpha_2)\in\mathbb{A}} \left(\begin{array}{l} \sum_{i=1}^{n} a_{ij}x_i \geq b_j, j \in J \\ x_1 = \alpha_1, x_2 = \alpha_2 \end{array} \right) \right\},$$

where $\mathbb{A} = \{(0,0),(0,1),(1,0),(1,1)\}$. Using Theorem 1, we derive a tight extended formulation for this set:

$$\sum_{i=1}^{n} a_{ij}\xi_i^h \geq b_j\xi_0^h, \quad j \in J \text{ and } h \in \{1,2,3,4\} \tag{12a}$$

$$\xi_1^1 = 0,\ \xi_2^1 = 0, \quad \xi_1^2 = \xi_0^2,\ \xi_2^2 = 0, \quad \xi_1^3 = 0,\ \xi_2^3 = \xi_0^3, \quad \xi_1^4 = \xi_0^4,\ \xi_2^4 = \xi_0^4, \tag{12b}$$

$$x_i = \xi_i^1 + \xi_i^2 + \xi_i^3 + \xi_i^4, \quad i \in \{1,\ldots,n\}, \tag{12c}$$

$$\xi_0^1 + \xi_0^2 + \xi_0^3 + \xi_0^4 = 1, \quad (\xi^h, \xi_0^h) \geq 0 \text{ for all } h \in \{1,2,3,4\}. \tag{12d}$$

Notice that $x_1 = \xi_0^2 + \xi_0^4$ and $x_2 = \xi_0^3 + \xi_0^4$. Now, assuming $\xi_0^1 = (1-x_1)(1-x_2)$, $\xi_0^2 = x_1(1-x_2)$, $\xi_0^3 = (1-x_1)x_2$, and $\xi_0^4 = x_1x_2$ ensure that constraints (12d) are satisfied. Also, for $i \in \{1,\ldots,n\}$, let $\xi_i^h = x_i\xi_0^h \geq 0$. It is easy to verify that constraints (12b) hold. Substitute $\{\xi_i^h\}_{i,h}$ in (12a) to get:

$$\left(\sum_{i=1}^{n} a_{ij}x_i \geq b_j\right)\xi_0^h, \quad j \in J \text{ and } h \in \{1,2,3,4\}. \tag{13}$$

Formulation (13) is same as multiplying each defining constraints with

$$\left(\prod_{k\in K_1} x_k\right)\left(\prod_{k\in K_2}(1-x_k)\right)$$

for each pair of disjoint sets (K_1, K_2) such $|K_1 \cup K_2| = 2$, i.e.,

$$(K_1,K_2) \in \big\{\, (\{1,2\},\emptyset)\,, (\emptyset,\{1,2\})\,, (\{1\},\{2\})\,, (\{2\},\{1\})\,\big\},$$

which is actually RLT-based reformulation of Sherali and Adams [2009].

Now for $p \geq 1$, set $\mathcal{X}^M$ in disjunctive normal form is given by

$$\bigvee_{(K_1,K_2)\in\mathbb{K}} \left(\begin{array}{c} \sum_{i=1}^{n} a_{ij}x_i \geq b_j, j \in J \\ x_k = 0 \ \forall k \in K_1, \quad x_k = 1 \ \forall k \in K_2 \end{array} \right),$$

where $\mathbb{K} := \{(K_1, K_2) : |K_1 \cup K_2| = p, K_1 \cap K_2 = \emptyset, K_1, K_2 \subseteq \{1, \ldots, p\}\}$. Again, using Theorem 1, we derive the following tight extended formulation for this set whose projection onto original space provides convex hull of $\mathcal{X}^M$:

$$\sum_{i=1}^{n} a_{ij}\xi_i^h \geq b_j\xi_0^h, \quad \text{for all } j \in J \text{ and } h \in \{1, \ldots, |\mathbb{K}|\}, \tag{14a}$$

$$\xi_k^h = 0, k \in K_1, \quad \xi_k^h = \xi_0^h, k \in K_2, \quad \text{for all } h \in \{1, \ldots, |\mathbb{K}|\}, \tag{14b}$$

$$x_i = \sum_{h=1}^{|\mathbb{K}|} \xi_i^h, \quad i \in \{1, \ldots, n\}, \tag{14c}$$

$$\sum_{h=1}^{|\mathbb{K}|} \xi_0^h = 1, \quad (\xi^h, \xi_0^h) \geq 0 \text{ for all } h \in \{1, \ldots, |\mathbb{K}|\}. \tag{14d}$$

By letting $\xi_i^h = x_i\xi_0^h$ for all $i \in \{1, \ldots, n\}$, and

$$\xi_0^h = \left(\prod_{k\in K_2} x_k \right) \left(\prod_{k\in K_1} (1 - x_k) \right),$$

constraints (14b)-(14d) are satisfied and constraints (14a) lead to the RLT-based reformulation of $\mathcal{X}^M$.

Remark 3 *As per our knowledge, this direct connection between RLT and linear programming equivalent of disjunctive programs without sequential convexification is not presented in the literature; Refer to Remark 1 of Sherali and Adams [2009] and Section 2.3 of Balas et al. [1993]. It implies that RLT-based reformulations can also be derived using Theorem 1 without the requirement of computing projections multiple times.*

Hierarchy of Relaxations for SIMBLPs. We present a hierarchy of relaxations between the continuous and convex hull representation of the set $\mathcal{X}^M$, i.e., $\mathcal{X}_{LP}^M$ and $conv(\mathcal{X}^M)$, respectively. A relaxation of $\mathcal{X}^M$ can be defined as

$$\mathcal{X}_t^R := \mathcal{X}_{LP}^M \cap \{x_i = 0 \vee x_i = 1, \text{ for } i = 1, \ldots, t\} \tag{15}$$

where $t \in \{1, \ldots, p\}$ and binary restrictions on $\{x_i\}_{i=t+1}^p$ variables have been relaxed. Clearly,

$$\mathcal{X}^M = \mathcal{X}_p^R \subseteq \mathcal{X}_{p-1}^R \subseteq \ldots \subseteq \mathcal{X}_1^R \subseteq \mathcal{X}_0^R = \mathcal{X}_{LP}^M,$$

and therefore, $conv(\mathcal{X}^M) \subseteq conv(\mathcal{X}^R_{t+1}) \subseteq conv(\mathcal{X}^R_t) \subseteq \mathcal{X}^M_{LP}$ for any $t \in \{1,\ldots,p-1\}$. This provides a hierarchy of relaxations for the feasible region of SIMBLPs. In the following proposition, we derive a tight extended formulation for $conv\left(\mathcal{X}^R_t\right)$, $t \in \{1,\ldots,p\}$.

Proposition 3 *For any $t \in \{1,\ldots,p\}$, the convex hull of $\mathcal{X}^R_t$ is equivalent to the projection of the following semi-infinite linear onto the original x-space:*

$$\sum_{i=1}^{n} a_{ij}\xi^h_i \geq b_j\xi^h_0, \quad \textit{for all } j \in J \textit{ and } h \in \{1,\ldots,|\mathbb{K}_t|\}, \tag{16a}$$

$$\xi^h_k = 0, k \in K_1, \quad \xi^h_k = \xi^h_0, k \in K_2, \quad \textit{for all } h \in \{1,\ldots,|\mathbb{K}_t|\}, \tag{16b}$$

$$x_i = \sum_{h=1}^{|\mathbb{K}_t|} \xi^h_i, \quad i \in \{1,\ldots,n\}, \tag{16c}$$

$$\sum_{h=1}^{|\mathbb{K}_t|} \xi^h_0 = 1, \quad (\xi^h, \xi^h_0) \geq 0 \textit{ for all } h \in \{1,\ldots,|\mathbb{K}_t|\}, \tag{16d}$$

where $\mathbb{K}_t := \{(K_1,K_2) : |K_1 \cup K_2| = t, K_1 \cap K_2 = \emptyset, K_1, K_2 \subseteq \{1,\ldots,t\}\}$.

Proof. Set $\mathcal{X}^R_t$ in the disjunctive normal form is written as

$$\bigvee_{(K_1,K_2)\in\mathbb{K}_t} \begin{pmatrix} \sum_{i=1}^{n} a_{ij}x_i \geq b_j, j \in J \\ x_k = 0 \ \forall k \in K_1, \quad x_k = 1 \ \forall k \in K_2 \end{pmatrix}.$$

Therefore, using Theorem 1, we derive the tight extended formulation (16) for this set whose projection onto original space provides convex hull of $\mathcal{X}^R_t$ for any $t \in \{1,\ldots,p\}$. □

Remark 4 *For finite binary linear programs, Sherali and Adams [1990] also present a hierarchy of relaxations using the RLT approach (Refer to Theorem 1 of Sherali and Adams [1990]). Interestingly, again by setting $\xi^h_i = x_i\xi^h_0$ for all $i \in \{1,\ldots,n\}$, and*

$$\xi^h_0 = \left(\prod_{k\in K_2} x_k\right)\left(\prod_{k\in K_1}(1-x_k)\right),$$

where $(K_1,K_2) \in \mathbb{K}_t$ in formulation (16) *with finite J, we can prove Theorem 1 of Sherali and Adams [1990]. This also showcase how linear programming equivalent for finite disjunctive programs by Balas [1979] is directly connected to hierarchy of relaxations provided by RLT in Sherali and Adams [1990].*

3.2 Partial Convex Hulls for SIMBLPs

Next, we utilize Theorem 1 to get a sequence of partial convex hulls for SIMBLP. Similar to the sequential convexification procedure of Balas et al. [1993] for SIMBLP with finite J, we multiply

each defining constraint of (11) with x_k and $(1 - x_k)$ for $k \in \{1, \ldots, p\}$ to get:

$$\sum_{i=1}^{n} a_{ij} x_i x_k \geq b_j x_k \qquad \text{and} \qquad \sum_{i=1}^{n} a_{ij} x_i (1 - x_k) \geq b_j (1 - x_k),$$

for all $j \in J$. By substituting $x_i x_k = y_i$ for $i \neq k$ and $x_k^2 = x_k$, the foregoing inequalities reduces to

$$\sum_{\substack{i=1 \\ i \neq k}}^{n} a_{ik} y_i + a_{kj} x_k \geq b_j x_k \quad \text{and} \quad \sum_{\substack{i=1 \\ i \neq k}}^{n} a_{ij} (x_i - y_i) \geq b_j (1 - x_k) \quad \text{for } j \in J. \tag{17}$$

Define $\mathcal{E}_k := \{(x, y) : (17) \text{ hold}\}$.

Theorem 2 *$Proj_{\mathbf{x}} \mathcal{E}_k = conv\left(\mathcal{X}_{LP}^M \cap \{x_k = 0 \vee x_k = 1\}\right)$.*

Proof. Observe that $\mathcal{X}_{LP}^M \cap \{x_k = 0 \vee x_k = 1\}$ is equivalent to

$$\begin{pmatrix} \sum_{i=1}^n a_{ij} x_i \geq b_j \text{ for all } j \in J \\ x_k = 0 \end{pmatrix} \bigvee \begin{pmatrix} \sum_{i=1}^n a_{ij} x_i \geq b_j \text{ for all } j \in J \\ x_k = 1 \end{pmatrix}.$$

Because of Theorem 1, we know that the convex hull of this disjunctive program with $H = \{1, 2\}$ is the projection of the following extended formulation onto the $\mathbf{x}$ space:

$$\sum_{i=1}^{n} a_{ij} \xi_i^h \geq b_j \xi_0^h, \quad j \in J \text{ and } h \in \{1, 2\}; \qquad \xi_k^1 = 0; \qquad \xi_k^2 = \xi_0^2; \tag{18a}$$

$$x_i = \xi_i^1 + \xi_i^2, \quad i \in \{1, \ldots, n\}; \qquad \xi_0^1 + \xi_0^2 = 1; \quad (\xi^1, \xi_0^1, \xi^2, \xi_0^2) \geq 0. \tag{18b}$$

From (18), it is easy to deduce that

$$\xi_k^1 = 0, \quad \xi_k^2 = \xi_0^2 = x_k, \quad \xi_0^1 = 1 - x_k. \tag{19}$$

By letting $\xi_i^2 = y_i$ and $\xi_i^1 = x_i - y_i$ for all $i \neq k$, and substituting $\xi_k^1, \xi_0^1, \xi_k^2, \xi_0^2$ from (19), the extended formulation (18) reduces to:

$$\sum_{\substack{i=1 \\ i \neq k}}^{n} a_{ij} (x_i - y_i) \geq b_j (1 - x_k), \quad j \in J \qquad \text{and} \qquad \sum_{\substack{i=1 \\ i \neq k}}^{n} a_{ij} y_i + a_{kj} x_k \geq b_j x_k, \quad j \in J,$$

which is same as $\mathcal{E}_k$. Hence, $\text{Proj}_{\mathbf{x}} \mathcal{E}_k = conv\left(\mathcal{X}_{LP}^M \cap \{x_k = 0 \vee x_k = 1\}\right)$. □

Remark 5 *This theorem generalizes Theorem 2.1 of Balas et al. [1993] for mixed binary programs. Moreover, $Proj_{\mathbf{x}}(\mathcal{E}_k)$ is referred to as a partial convex hull of $\mathcal{X}^M$. In other words,*

$$\mathcal{X}^M \subseteq Proj_{\mathbf{x}}(\mathcal{E}_k) \text{ and } conv\left(Proj_{\mathbf{x}}(\mathcal{E}_k) \cap \{x_i \in \{0, 1\} : i \neq k\}_{i=1}^p\right) = conv(\mathcal{X}^M),$$

where $Proj_{\mathbf{x}}(\mathcal{E}_k)$ *has* $p-1$ *binary variables.*

Definition 2 (Sequential Convexification) *Let* $\mathfrak{P}_k(\mathcal{X}_{LP}^M) = Proj_{\mathbf{x}}(\mathcal{E}_k)$. *Given a sequence* $\{k_1, \ldots, k_t\} \subseteq \{1, \ldots, p\}$, *define*

$$\mathfrak{P}_{k_1,\ldots,k_t}\left(\mathcal{X}_{LP}^M\right) = \mathfrak{P}_{k_t}\left(\ldots\left(\mathfrak{P}_{k_2}\left(\mathfrak{P}_{k_1}\left(\mathcal{X}_{LP}^M\right)\right)\right)\ldots\right).$$

The following result proves that $\mathfrak{P}_{k_1,\ldots,k_t}\left(\mathcal{X}_{LP}^M\right)$ is a partial convex hull of $\mathcal{X}^M$ with no binary restrictions on $x_{k_1}, \ldots, x_{k_t}$ variables. Our proof for this theorem relies on Theorem 2 and facial property of $\mathcal{X}^M$, i.e., $x_k = 0$ and $x_k = 1$, $k \in \{1, \ldots, p\}$, are faces of $\mathcal{X}_{LP}^M$.

Theorem 3 *Given any sequence* $\{k_1, \ldots k_t\} \subseteq \{1, \ldots, p\}$,

$$\mathfrak{P}_{k_1,\ldots,k_t}\left(\mathcal{X}_{LP}^M\right) = conv\left(\mathcal{X}_{LP}^M \cap \left\{x_k = 0 \vee x_k = 1\right\}_{k=k_1}^{k_t}\right).$$

Proof. The arguments of induction-based proof for Theorem 2.2 in Balas et al. [1993] are applicable to prove this result as well if we use Theorem 2 of this paper, instead of Theorem 2.1 of Balas et al. [1993]. □

Corollary 1 *For* $\{k_1, \ldots k_p\} = \{1, \ldots, p\}$, $\mathfrak{P}_{k_1,\ldots,k_p}\left(\mathcal{X}_{LP}^M\right) = conv(\mathcal{X}^M)$.

3.3 Solution Approaches for SIMBLPs

We provide a sequential convexification approach and a cutting plane algorithm for solving SIMBLPs. These approaches extend and are build upon sequential convexification and a lift-and-project algorithm of Balas et al. [1993] for MBLPs. Given a fractional solution $\widehat{\mathbf{x}}$ where $0 < x_k < 1$ for $k \in \{1, \ldots, p\}$, we construct extended formulation and cutting planes using Theorem 3 and Proposition 2, respectively, to cut-off the point $\widehat{\mathbf{x}}$.

Sequential Convexification: Let $\mathbf{x}^0$ be an optimal solution corresponding to $\mathcal{X}_{LP}^M$ and r denotes iteration counter. In iteration r, let $\mathcal{E}^r$ be an extended formulation (a semi-infinite linear program) in $(\mathbf{x}, \mu^r)$ space such that $\mathcal{X}_{LP}^M = \mathcal{E}^0 \supseteq \text{Proj}_{\mathbf{x}}(\mathcal{E}^{r-1}) \supseteq \text{Proj}_{\mathbf{x}}(\mathcal{E}^r) \supseteq \mathcal{X}^M$. Let $(\widehat{\mathbf{x}}^r, \widehat{\mu}^r)$ be an optimal solution associated with $\mathcal{E}^r$ where $\widehat{x}_k$ is fractional. Using Theorem 2, we derive an extended formulation $\mathcal{E}^{r+1}$ such that $\text{Proj}_{\mathbf{x}}(\mathcal{E}^{r+1}) = conv(\mathcal{E}^r \cap (x_k = 0 \vee x_k = 1))$. This process is repeated until an integral solution is obtained. Because of Theorem 3, it terminates in at most p iterations.

Cutting Plane Algorithm with Cuts in Original Space: We derive valid inequalities for SIMBLPs in the original space using Proposition 2, and embed them within an iterative process to get a cutting plane algorithm for SIMBLPs. Let $\mathcal{X}_{relax}^M = \mathcal{X}_{LP}^M \cap \{\mathbf{x} : \sum_{i=1}^n a_{ij}x_i \geq b_j \text{ for all } j \in J_C\}$ be a relaxation of $\mathcal{X}^M$ where $\sum_{j=1}^n a_{ij}x_{ij} \geq b_j$ for $j \in J_C$, is a valid inequality for $\mathcal{X}^M$. Also, let $\widehat{\mathbf{x}}$ be an optimal solution associated to this relaxation such that $x_k \in (0, 1)$ for some $k \in \{1, \ldots, n\}$.

Proposition 4 *Assuming that the set of points* $\{(a_j, b_j) \in \mathbb{R}^{n+1} : j \in J \cup J_C\}$ *is closed and bounded. An inequality* $\pi^T x \geq \pi_0$ *is valid for* $conv(\mathcal{X}^M)$ *if and only if there exists vectors* $(\sigma^h, \sigma_0^h) \in \mathbb{R}_+^{m+1}$

for $h \in \{1,2\}$ such that for any $k \in \{1,\ldots,n\}$,

$$\pi_i \geq \sum_{j \in J_m^h} \sigma_j^h a_{ij} \quad \textit{for } i \in \{1,\ldots,n\}\backslash\{k\} \textit{ and } h = 1,2; \tag{20a}$$

$$\pi_k \geq \sum_{j \in J_m^1} \sigma_j^1 a_{kj} - \sigma_0^1, \qquad \pi_k \geq \sum_{j \in J_m^2} \sigma_j^2 a_{kj} + \sigma_0^2, \tag{20b}$$

$$\pi_0 \leq \sum_{j \in J_m^1} \sigma_j^1 b_j, \qquad \pi_0 \leq \sum_{j \in J_m^2} \sigma_j^2 b_j + \sigma_0^2, \tag{20c}$$

where $J_m^h \subseteq J \cup J_C$ and $|J_m^h| = m \leq n$.

Proof. For any $k \in \{1,\ldots,n\}$, $\mathcal{X}^{M,k} = \mathcal{X}^M_{relax} \cap \{(x_k \leq 0) \vee (x_k \geq 1)\}$ subsumes $\mathcal{X}^M$. Therefore, any valid inequality for $\mathcal{X}^{M,k}$ is also valid for $\mathcal{X}^M$. Recall that $\mathcal{X}^{M,k}$ is equivalent to

$$\begin{pmatrix} \sum_{i=1}^n a_{ij}x_i \geq b_j \text{ for all } j \in J \cup J_C \\ -x_k \geq 0 \end{pmatrix} \bigvee \begin{pmatrix} \sum_{i=1}^n a_{ij}x_i \geq b_j \text{ for all } j \in J \cup J_C \\ x_k \geq 1 \end{pmatrix}.$$

By applying Proposition 2, we get conditions (20) for which $\pi^T x \geq \pi_0$ is valid for $\mathcal{X}^{M,k}$ and, hence valid for $conv(\mathcal{X}^M)$ as well. □

Note that wlog, we assume that $\mathcal{X}^M_{LP} \cap \{x_i = \alpha\}$ is nonempty for any $i \in \{1,\ldots,n\}$ and $\alpha \in \{0,1\}$. Now, using Proposition 4, a valid inequality $\pi^T x \geq \pi_0$ is generated for the point $\widehat{\mathbf{x}}$ by solving the following program: $\min\{\pi^T\widehat{\mathbf{x}} - \pi_0 : (\pi,\pi_0) \text{ satisfies (20)}\}$. To get a pure cutting plane algorithm for SIMBLPs, we incorporate the foregoing cut-generation procedure within the specialized lift-and-project approach for MBLPs (refer to Page 309 of Balas et al. [1993] for details).

Remark 6 *We can also embed the aforementioned approaches for SIMBLPs within the branch-and-cut framework introduced in Section 4. In other words, at each non-leaf node with fractional solution, cuts and tighter extended formulations approximations are generated for the node subproblem to get stronger lower bounds.*

3.4 Extensions and Applications of SIMBLPs and SIDPs

Various mathematical optimization programs can be reformulated as or are special cases of SIMBLPs or SIDPs, respectively. For an example, disjunctive programs subsume nonconvex quadratic programs, separable non-linear programs, and many more (refer to Balas [1998] for details). Similarly, Sherali and Adams [2009] showed how the aforementioned RLT-based reformulation for SIMBLPs can also be utilized for convex mixed-binary programs, semi-infinite mixed-discrete programs, and convex mixed-discrete programs. They redefined the latter programs as SIMBLPs. As a consequence, Theorem 3 is applicable to sequentially convexify these programs as well. In this section, we discuss applications of SIDPs and SIMBLPs for solving disjunctive convex programs, SIMBLPs with discrete variables, robust 0-1 knapsack problems, distributionally robust programs, and distributionally robust chance-constraint programs.

(*i*) *Semi-Infinite Mixed-Discrete Linear Programs.* This class of problem is defined over set:

$$\left\{\mathbf{x} \in \mathbb{R}^n_+ : \left(\sum_{i=1}^{n} a_{ij}x_i \geq b_j, j \in J\right) \wedge \left(\bigvee_{l=1}^{m}(x_i = \alpha_l), i = 1, \ldots, p\right)\right\}, \tag{21}$$

where $\alpha_i \in \mathbb{R}_+$ and m is finite. Moreover, the defining constraints include $x_i \leq 1$ for $i = p{+}1, \ldots, n$. Using auxiliary binary variables $\nu_{il} = 1$, if $x_i = \alpha_l$ and 0 otherwise, the disjunctive constraints in set (21) reduces to:

$$\left(x_i = \sum_{l=1}^{m} \alpha_l \nu_{il} \text{ for } i = 1, \ldots, p; \quad \sum_{l=1}^{m} \nu_{il} = 1 \text{ for } i = 1, \ldots, p\right), \tag{22}$$

and hence these programs (21) belong to the class of SIMBLPs.

(*ii*) *Disjunctive Convex Programs* are defined by disjunction of a finite number of convex sets, i.e.,

$$\mathcal{D}_C := \left\{\mathbf{x} \in \mathbb{R}^n : \bigvee_{h \in H} \mathcal{D}^h_C\right\} \text{ where } \mathcal{D}^h_C := \left\{\mathbf{x} \in \mathbb{R}^n_+ : f^h_k(\mathbf{x}) \leq 0, k \in K\right\}$$

for $h \in H$, is a nonempty and bounded convex set, and $f^h_k(\cdot)$ is a differentiable convex function. Using standard first-order approximation of the convex functions, $f^h_k(\mathbf{x}) \leq 0$ can be written as a semi-infinite linear program, i.e., $\nabla f^h_k(\widehat{\mathbf{x}}^k)^T(\mathbf{x} - \widehat{\mathbf{x}}^k) \leq 0$ for all (boundary points) $\widehat{\mathbf{x}}^k \in \mathcal{B}_{hk} := \{\mathbf{x} \in \mathbb{R}^n_+ : f^h_k(\mathbf{x}) = 0\}$. Now, according to Theorem 1, a tight extended formulation of $\mathcal{D}_C$ is given by:

$$\mathbf{x} = \sum_{h \in H} \zeta^h, \tag{23a}$$

$$\nabla f^h_k(\widehat{\mathbf{x}}^k)^T(\zeta^h - \widehat{\mathbf{x}}^k\zeta^h_0) \leq 0, \quad \widehat{\mathbf{x}}^k \in \mathcal{B}_{hk},\ h \in H, \text{ and } k \in K, \tag{23b}$$

$$\sum_{h \in H} \zeta^h_0 = 1, \tag{23c}$$

$$(\zeta^h, \zeta^h_0) \geq 0, \quad h \in H. \tag{23d}$$

Since D^h_C is a bounded set, there does not exist any direction/vector such that $f^h_k(\mathbf{x})^T\mathbf{d} \leq 0$, unless $\mathbf{d} = 0$. As a result, $\zeta^h_0 = 0$ in (23b) implies that $\zeta^h = 0$. Therefore, constraints (23b) reduces to $f^h_k(\zeta^h/\zeta^h_0) \leq 0$ for all $h \in H$, $k \in K$ when $\zeta^h_0 > 0$, and trivial constraint $(0 \leq 0)$, otherwise.

(*iii*) *Robust 0-1 Knapsack Problem.* The well-known 0-1 knapsack problem is defined as follows:

$$\min_{\mathbf{x} \in \{0,1\}^n} \left\{\sum_{i=1}^{n} c_i x_i : \sum_{i=1}^{n} \alpha_i x_i \geq \beta\right\}.$$

Assume that the coefficients (α, β) are uncertain and they belong to a so-called uncertainty set Ξ (polytope or ellipsoidal set). Then, the robust 0-1 knapsack problem is given by

$$\min_{\mathbf{x}\in\{0,1\}^n} \left\{ \sum_{i=1}^{n} c_i x_i : \sum_{i=1}^{n} \alpha_i x_i \geq \beta \text{ for all } (\alpha,\beta) \in \Xi \right\}.$$

Notice that this problem is equivalent to SIBLP.

(iv) *Distributionally Robust Programs.* Given a set of distributions $\mathfrak{P}$, referred to as ambiguity set, a distributionally robust program is defined as

$$\min_{\mathbf{x}\in\mathcal{X}_D} \max_{P\in\mathfrak{P}} \mathbb{E}_P[H(x,\rho)], \tag{DRO}$$

where $\mathcal{X}_D \subseteq \mathbb{R}^n_+$, ρ is a random variable with P probability distribution and Ω sample space (a set of realizations of ρ), and $\mathbb{E}_P[\cdot]$ is an expectation operator. Assuming that $|\Omega|$ is finite, (DRO) can also be written as

$$\min_{\mathbf{x}\in\mathcal{X}_D} \left\{ \Theta : \sum_{\omega\in\Omega} p_\omega H(x,\omega) \leq \Theta, \quad \{p_\omega\}_{\omega\in\Omega} \in \mathfrak{P} \right\}. \tag{DRO-F}$$

Depending on definition of $\mathcal{X}_D$ and $H(x,.)$, (DRO-F) translates to variety of semi-infinite programs. Here are a few examples: Let $H(x,.)$ be maximization over a finite set of piecewise linear function. When $\mathcal{X}_D$ is a disjunctive set, mixed-discrete set, or mixed binary set, it is equivalent to SIDP, semi-infinite mixed-discrete linear program (21), or SIMBLP, respectively.

(v) *Distributionally Robust Chance-Constraint Problems.* Given a system of linear inequalities $T\mathbf{x} \geq \rho$ where ρ is a random variable having a finite set of realizations $\{\rho_1, \ldots, \rho_M\}$ with probability p_ω for $\rho = \rho_\omega$, $\omega \in \{1, \ldots, M\}$. A probabilistic (or chance) constraint is denoted by $P(T\mathbf{x} \geq \rho) \leq \epsilon$ where $P = \{p_\omega\}_{\omega\in\Omega}$ and ϵ is known. Accordingly, a distributionally robust chance-constraint problem is defined as follows:

$$\min_{\mathbf{x}\in\mathcal{X}_C\subseteq\mathbb{R}^n_+} \left\{ \sum_{i=1}^{n} c_i x_i : \sup_{P\in\mathfrak{P}} \left\{ P(T\mathbf{x} \geq \rho) \right\} \leq \epsilon \equiv P(T\mathbf{x} \geq \rho) \leq \epsilon \ \text{ for } P \in \mathfrak{P} \right\}. \tag{DRO-CC}$$

Let $z_\omega = 0$, if $\rho = \rho_\omega$ and 1, otherwise. Asuming $T\mathbf{x} \geq 0$, the feasible region of (DRO-CC) reduces to

$$\left\{ (\mathbf{x}, z) \in \mathcal{X}_C \times \{0,1\}^M : T\mathbf{x} + \rho_\omega z_\omega \geq \rho_\omega \ \text{ for } \omega \in \{1,\ldots,M\} \right.$$
$$\left. \sum_{\omega=1}^{M} p_\omega (1 - z_\omega) \leq \epsilon \ \text{ for all } \{p_\omega\}_\omega \in \mathfrak{P} \right\},$$

which belongs to the class of SIMBPs when $\mathcal{X}_C = \mathbb{R}^n_+$.

4 Algorithm for Semi-Infinite Binary Linear Programs

A semi-infinite binary linear program (SIBLP) is defined as:

$$\min_{\mathbf{x}\in\{0,1\}^n} \left\{ \sum_{i=1}^{n} c_i x_i : \sum_{i=1}^{n} a_{ij} x_i \geq b_j \text{ for all } j \in J \right\}, \tag{SIBLP}$$

where $|J|$ is not finite. In this section, we study SIBLPs with implicity defined constraints (SIBLP-IC) whose constraints are not explicity known but given a point $\widehat{\mathbf{x}} \in \mathbb{R}^n$, there exists an oracle to either certify

$$\sum_{i=1}^{n} a_{ij}\widehat{x}_i - b_j \geq 0 \;\; \text{for all } j \in J,$$

(all constraints are satisfied) or return $\tau \in J$ such that $\sum_{i=1}^{n} a_{i\tau}\widehat{x}_i - b_\tau < 0$ (violated constraint). Using this constraint-generation oracle (C-GO), we present a branch-and-cut based approach for solving SIBLPs. Note that Gustafson and Kortanek [1973] and Gribik [1979] introduced algorithms for linear programming relaxation of SIBLP assuming that this oracle exists.

Observation 1 *Let $J_0 \subseteq J$ be a subset of constraints derived using C-GO. Also, we define:*

$$\mathcal{X}(J_0) := \left\{ \mathbf{x} \in \{0,1\}^n : \sum_{i=1}^{n} a_{ij} x_i \geq b_j, \;\; j \in J_0 \right\},$$

and $\mathcal{O}(J_0) := \min\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in \mathcal{X}(J_0)\}$. Then, the following statements hold:

(a) $\mathcal{O}(J_0) \leq \mathcal{O}(J)$ and $\mathcal{X}(J) \subseteq \mathcal{X}(J_0)$ for all $J_0 \subseteq J$;

(b) For $\widehat{\mathbf{x}} \in \mathcal{X}(J_0) \setminus \mathcal{X}(J)$, $\mathbf{c}^T\widehat{\mathbf{x}} \leq \mathcal{O}(J)$;

(c) For $\widehat{\mathbf{x}} \in \mathcal{X}(J_0) \cap \mathcal{X}(J)$, $\mathbf{c}^T\widehat{\mathbf{x}} \geq \mathcal{O}(J)$.

Below we introduce steps of the branch-and-cut approach for SIBLPs where $UB^{orig} \leftarrow \infty$ and $LB^{orig} \leftarrow -\infty$ denote upper and lower bounds, respectively, for $\mathcal{O}(J)$, i.e., optimal solution value of SIBLP. We also denote (SIBLP) by SIBLP(J) and its relaxations by SIBLP(J_0) for $J_0 \subseteq J$.

Step I (Root Node): At the root node of the branch-and-cut tree, we derive an initial subset of constraints $J_0 \subset J$ using the C-GO. Specifically, let $\mathbf{x}_0 \leftarrow 0$ and $J_0 \leftarrow \emptyset$ be initialization vector and subset, respectively. We call C-GO to get a constraint (index) $\tau \in J$ violated by $\mathbf{x}_0$. We update $J_0 \leftarrow J_0 \cup \{\tau\}$, solve linear programming relaxation of SIBLP(J_0), and use its optimal solution as a new initialization vector $\mathbf{x}_0$. This process is repeated a pre-determined finite number of times, thereby leading to a finite set $J_0 \subset J$ and a relaxation SIBLP(J_0). Since $\mathbf{x}_0$ is an optimal solution of LP relaxation of SIBLP(J_0), $\mathbf{c}^T\mathbf{x}_0$ is a lower bound for both $\mathcal{O}(J_0)$ and $\mathcal{O}(J)$. We denote upper and lower bounds of optimal solution value of this relaxation $\mathcal{O}(J_0)$ by UB^{relax} and LB^{relax}, respectively.

Step II (Branching): At each node, we solve linear programming relaxation of SIBLP(J_0) with fixed values of a subset of the binary variables. Let LB_{node} and $\widehat{\mathbf{x}}_{node}$ denote the optimal solution value and an optimal solution, respectively, for the node subproblem. Similar to generic branch-and-bound approach for 0-1 programs [Nemhauser and Wolsey, 1988], we create branches of the tree depending on fractional components of $\widehat{\mathbf{x}}_{node}$, as if we are solving SIBLP(J_0).

Step III (Cut-Generation): At a leaf node of the tree, a feasible solution $\widehat{\mathbf{x}}_{ln} \in \mathcal{X}(J_0)$ of SIBLP(J_0) is obtained, and upper bound of $\mathcal{O}(J_0)$, denoted by $UB^{relax} \leftarrow \min\{UB^{relax}, \mathbf{c}^T\widehat{\mathbf{x}}_{ln}\}$, is updated. By calling C-GO, we either

1. *Certify that it is a feasible solution of SIBLP(J) as well*: This implies that $\mathbf{c}^T\widehat{\mathbf{x}}_{ln}$ is an upper bound for $\mathcal{O}(J)$, and therefore, we update $UB^{orig} \leftarrow \min\{UB^{orig}, \mathbf{c}^T\widehat{\mathbf{x}}_{ln}\}$ if needed; or

2. *Get a constraint $\tau \in J \setminus J_0$ violated by $\widehat{\mathbf{x}}_{ln}$:* This implies that $\widehat{\mathbf{x}}_{ln} \in \mathcal{X}(J_0) \setminus \mathcal{X}(J)$, and therefore, $\mathbf{c}^T\widehat{\mathbf{x}}_{ln}$ is a lower bound of $\mathcal{O}(J)$. We update $LB^{orig} \leftarrow \max\{LB^{orig}, \mathbf{c}^T\widehat{\mathbf{x}}_{ln}\}$. Additionally, we append $J_0 \leftarrow J_0 \cup \{\tau\}$, solve linear programming relaxation of SIBLP(J_0) to get an optimal solution $\widehat{\mathbf{x}}^{LP}_{ln} \in [0,1]^n$, and call C-GO to get another constraint, if any, violated by $\widehat{\mathbf{x}}^{LP}_{ln}$. The forgoing process is repeated a pre-determined number of times. We use the appended J_0 for the rest of the unexplored tree.

Apart from the defining constraints, other cutting planes (discussed in previous sections) can also be added at each node.

Step IV (Pruning): A node is pruned in either of the following cases:

(i) If $LB_{node} \geq UB^{orig}$, then we prune this node. It is important to note that in usual branch-and-bound tree for SIBLP(J_0), a node is pruned when $LB_{node} \geq UB^{relax}$. However, since $UB^{relax} \leq UB^{orig}$, the pruning condition $LB_{node} \geq UB^{relax}$ could lead to neglecting exploration of optimal solution(s) of SIBLP(J).

(ii) If $\widehat{\mathbf{x}}_{node}$ is integral, then it is feasible for SIBLP(J_0). Usually such nodes are pruned after updating $UB^{relax} \leftarrow \min\{UB^{relax}, \mathbf{c}^T\widehat{\mathbf{x}}_{node}\}$. However, we have to first call the C-GO to ensure whether it is a feasible solution of SIBLP(J) or not. If yes, we update $UB^{orig} \leftarrow \min\{UB^{orig}, \mathbf{c}^T\widehat{\mathbf{x}}_{node}\}$ and prune the node. Otherwise, we update $LB^{orig} \leftarrow \max\{LB^{orig}, \mathbf{c}^T\widehat{x}_{node}\}$ and append J_0 as discussed above, and then prune the node.

(iii) In case $\widehat{\mathbf{x}}_{node}$ does not exist.

Step V (Optimality/Termination): The algorithm terminates after exploring all nodes of the tree and returns the best known upper bound UB^{orig} as the optimal solution value, along with associated optimal solution.

Proposition 5 *The branch-and-cut approach for SIBLP converges to an optimal solution in finite*

iterations.

Proof. Let $\{J_0^1, J_0^2, \ldots\}$ be a sequence of subsets of J generated during this algorithm such that $J_0 = J_0^1 \subset J_0^2 \subset \ldots \subset J$. The convergence proof follows from the finiteness of binary solutions in $\mathcal{X}(J_0^1) \supseteq \mathcal{X}(J_0^2) \supseteq \ldots \supseteq \mathcal{X}(J)$ that are implicitly enumerated through this approach. □

4.1 Application to Solve Implicit Hitting Set and Set-Covering Problems

As discussed in Section 1, binary formulation for set-covering problem with infinite demand points and implicit hitting set problem (IHSP) are special cases of SIBLP-IC. In Moreno-Centeno and Karp [2013], the authors presented around 12 combinatorial optimization problems that can be reformulated as IHSP. It includes multigenome alignment, max cut, matroid intersection, max 2-SAT, (un)directed traveling salesperson problem, minimum feedback vertex/arc set problems, among others. We present the exact algorithm of Moreno-Centeno and Karp [2013] for IHSP, denoted by `Exact-IHSP`, so that a reader can establish distinction between this approach and the algorithm proposed in the previous section for general SIBLP-IC.

Algorithm for IHSP [Moreno-Centeno and Karp, 2013]. We denote a relaxation of (2) with $\mathcal{S}_k \subseteq \mathcal{S}$ constraints by IHSP($\mathcal{S}_k$). When set $\mathcal{S}_k$ is explicitly known, this relaxation is an Explicit Hitting-Set Problem (EHSP) that is an NP-complete problem [Karp, 1972]. However, a feasible solution of EHSP can be obtained using a polynomial-time greedy heuristic. The C-GO for IHSP is an oracle that given a solution $\widehat{\mathbf{x}} \in \{0,1\}^n$, ensures either all constraints in set $\mathcal{S}$ are satisfied by this solution, i.e., $\sum_{i \in C} \hat{x}_i \geq 1$ for all $C \in \mathcal{S}$, or it returns a subset of the constraints that are violated by $\hat{\mathbf{x}}$. Pseudocode of `Exact-IHSP` is given in Algorithm 4.1 and it works as follows. Initialize by setting best upper bound UB to ∞, and iteration counter $L = 1$. Let $\widehat{\mathbf{x}} \in \{0,1\}^n$ be an initial vector. Using the C-GO for $\widehat{\mathbf{x}}$, derive an EHSP relaxation of IHSP, i.e., a finite set $\mathcal{S}_L \subseteq \mathcal{S}$, such that a feasible solution of IHSP($\mathcal{S}_L$) provided by the greedy approach is also feasible for IHSP (2), denoted by $\overline{\mathbf{x}}_L$. It is important to note that the set of feasible solution of EHSP subsumes the feasible solutions of IHSP. Therefore, a feasible solution of EHSP can either be a feasible solution of IHSP or it provides a lower bound for optimal solution value of IHSP. Since solution value associated with $\overline{\mathbf{x}}_L$, i.e., $\mathbf{1}\overline{\mathbf{x}}_L$, is an upper bound for (2), we update UB if needed. We also solve IHSP($\mathcal{S}_L$) to optimality to get an optimal solution $\widehat{\mathbf{x}}_L$ and optimal solution value $\mathbf{1}\widehat{\mathbf{x}}_L$, which is a lower bound LB for IHSP (2). In case, $LB = UB$ or $\widehat{\mathbf{x}}_L$ is a feasible solution of (2), we terminate the algorithm with $\widehat{\mathbf{x}}_L$ as an optimal solution. Otherwise, we set $\widehat{\mathbf{x}} \leftarrow \widehat{\mathbf{x}}_L$, increment the iteration counter $L \leftarrow L + 1$, and repeat the aforementioned steps (Steps 2-5 in Algorithm 4.1) until the algorithm terminate. Moreno-Centeno and Karp [2013] proved that this algorithm converges to an optimal solution of IHSP.

One of the key features of this algorithm is the utilization of a polynomial time greedy method for a relaxation of (2) to get a feasible solution for (2). Interestingly, this algorithm can also be viewed as a cutting-plane approach with the foregoing feature. Specifically, given a binary program, it performs the following steps iteratively: (a) Derive a relaxation with a subset of the

Algorithm 1 Exact Algorithm for IHSP

1: Let $UB \leftarrow \infty$, iteration counter $L \leftarrow 1$, and initial vector $\widehat{\mathbf{x}} \in \{0,1\}^n$;
2: Repeatedly derive $\mathcal{S}_L \subset \mathcal{S}$ using C-GO and $\widehat{\mathbf{x}}$, until a feasible solution $\overline{\mathbf{x}}_L$ of IHSP($\mathcal{S}_L$) provided by greedy heuristic is feasible for IHSP($\mathcal{S}$); `Refer to Moreno-Centeno and Karp [2013] for details.`
3: Update upper bound $UB \leftarrow \min\{UB, \mathbf{1}\overline{\mathbf{x}}_L\}$;
4: Let $\widehat{\mathbf{x}}_L$ and LB be an optimal solution and the optimal solution value of IHSP($\mathcal{S}_L$);
5: **if** $LB = UB$ or $\widehat{\mathbf{x}}_L$ is a feasible solution of IHSP($\mathcal{S}$) **then** Return $\widehat{\mathbf{x}}_L$;
6: **else** Update $\widehat{\mathbf{x}} \leftarrow \widehat{\mathbf{x}}_L$ and $L \leftarrow L + 1$, and Go to Step 2;
7: **end if**

defining constraints; (b) Obtain a feasible solution of the relaxation such that it is feasible for the original problem as well, thereby provide an upper bound; (c) Solve the relaxation to optimality to get a lower bound; (d) Terminate if the optimal solution of the relaxation is feasible for the original problem or optimality gap (difference between upper and lower bounds) is negligible. Notice that this approach is not applicable for general SIBLP-IC and its other extensions/variants considered in this paper. Additionally, it does not incorporate efficient methods available for solving EHSP or IHSP($\mathcal{S}_L$) without binary restrictions using methods of Koufogiannakis and Young [2007], Plotkin et al. [1995] that can be used at each node of the branch-and-cut algorithm for SIBLP-IC.

4.2 Computational Experiments

In this section, we present results of our computational experiments conducted on instances of set-covering problem with infinite number of demand points. The purpose of these experiments is to evaluate performance and effectiveness of algorithm presented in Section 4 (denoted by `Exact-SIBLP`) for solving SIBLP-IC, in comparison to solving this formulation directly using Gurobi and its implicit hitting set equivalent using `Exact-IHSP`. Since Gurobi does not allow infinite number of constraints, we randomly generate instances with a large number of demand points $|\mathcal{S}| \in \{100000, 500000\}$. This is done as follows. Let the number of subsets (or potential facility locations) $m \in \{500, 750, 1000, 1250\}$. We generate demand points and randomly assign each of them to at least one of the m subsets. The number of subsets containing a demand point is a positive integer number drawn from Uniform[1,m]. We coded `Exact-SIBLP` and `Exact-IHSP` using Python 3.9.13 and Gurobi 9.5.2 with default settings, and conducted experiments on a laptop with 11th Gen Intel(R) Core(TM) i7 3GHz processor, 16GB RAM, and Windows 10 Enterprise.

We also design a simple/naive heuristic as C-GO for solving set-covering problem using `Exact-SIBLP` where in each call no more than 1000 cuts are generated. For a given binary solution, we enumerate each demand point that is not included in the initial subset J_0 until either it is verified that all demand points are covered or the number of newly generated cuts is at most 1000. For realistic data points distributed over a two-dimensional plane and subsets defined by spatial objects, a geographic information system (GIS) embedded with some machine learning techniques can also be used as C-GO to determine demand points not covered by selected subsets [Murray, 2010].

In Tables 1-3, we present the results for our experiments. Each row of these tables represents an instance of the set-covering problem with m subsets and $|\mathcal{S}|$ number of demand points that are assumed to be not known explicitly for `Exact-SIBLP` and Exact-IHSP. For `Exact-SIBLP`, we report number of nodes (#Nodes), number of times C-GO is called (#C-GO), number of cuts

Inst#	m	$\lvert\mathcal{S}\rvert$	OPT	`Exact-SIBLP`						
				#Nodes	#CG-O	#Cuts	CG-Time	BnB-Time	T-Time	$\lvert J_0\rvert$
I.1	500	5×10^4	181	2	302	23	1.43	0.42	1.85	500
				1	336	6	0.92	0.80	1.72	5000
				11	289	186	0.29	3.19	3.48	25000
I.2	500	5×10^4	178	87	535	30	1.35	1.23	2.58	500
				131	630	28	1.13	1.02	2.15	5000
				159	573	184	0.55	3.4	3.95	25000
I.3	500	5×10^4	183	47	492	30	1.82	0.74	2.56	500
				11	473	0	1.46	1.08	2.54	5000
				357	1035	203	0.59	3.44	4.03	25000
II.1	500	10^5	259	1	266	12	1.59	0.30	1.89	1000
				1	203	9	1.87	1.03	2.90	10000
				1	222	227	0.37	4.68	5.05	50000
II.2	500	10^5	267	1	304	5	1.79	0.73	2.52	1000
				1	258	0	1.87	1.10	2.97	10000
				1	267	3	0.93	7.46	8.39	50000
II.3	500	10^5	224	17	418	0	2.10	1.17	3.27	1000
				159	647	72	1.75	1.32	3.07	10000
				71	571	4	0.93	7.83	8.76	50000
III.1	750	10^5	258	107	654	30	3.14	1.80	4.94	1000
				71	506	28	2.38	2.25	4.63	10000
				223	803	287	1.48	11.66	13.14	50000
III.2	750	10^5	250	71	542	44	3.55	1.56	5.11	1000
				191	867	104	3.68	3.26	6.94	10000
				537	1361	308	0.77	11.90	12.67	50000
III.3	750	10^5	258	323	997	79	3.47	1.89	5.36	1000
				390	1147	39	2.33	2.59	4.92	10000
				997	2339	286	0.76	12.51	13.27	50000
IV.1	750	5×10^5	538	1	293	9	4.57	1.09	5.66	5000
				1	243	1408	2.72	10.88	13.60	50000
				1	259	407	2.07	88.69	90.76	250000
IV.2	750	5×10^5	530	1	258	2142	2.66	0.71	3.37	5000
				1	242	1394	2.18	7.82	10.00	50000
				1	240	316	1.74	59.58	61.32	250000
IV.3	750	5×10^5	536	1	289	14	2.27	1.09	3.36	5000
				1	245	1502	2.16	7.02	9.18	50000
				1	263	372	2.06	53.60	55.66	250000

Table 1: Computational results for `Exact-SIBLP` using different J_0 set and $m \in \{500, 750\}$

added by C-GO (#Cuts), total time taken by C-GO (CG-Time), total branch-and-bound time excluding C-GO time (BnB-Time), total solution time (T-Time = CG-Time + BnB-Time) and optimal solution value (OPT). Similarly, for Gurobi and `Exact-IHSP`, we report solution time and number of nodes. We also evaluate the performance of `Exact-SIBLP` for different sizes of initial subset J_0, and present results in Tables 1 and 2 for $|J_0|/|\mathcal{S}| \in \{0.01, 0.1, 0.5\}$. In Table 3, we include the results of `Exact-SIBLP` for subset J_0 that takes minimum total time (CG-Time plus BnB-Time). We set a time limit of 3600 seconds for each experiment and in all tables, the runtimes are reported in seconds. For instances that were not solved to optimality within the time limit, we report 3600+ as solution time and OPT provides best known feasible solution with the remaining

Inst#	m	$\lvert\mathcal{S}\rvert$	OPT	`Exact-SIBLP`						
				#Nodes	#CG-O	#Cuts	CG-Time	BnB-Time	T-Time	$\lvert J_0\rvert$
V.1	1000	10^5	251	6532	13982	93	2.68	15.35	18.03	1000
				8732	18210	22	3.21	18.04	21.25	10000
				3993	8558	0	1.04	15.25	16.29	50000
V.2	1000	10^5	237	23665	49007	100	5.09	48.62	53.71	1000
				32498	67140	138	5.05	94.10	99.15	10000
				14437	30392	32	3.52	106.87	110.39	50000
V.3	1000	10^5	255	2322	6275	33	12.23	40.97	53.20	1000
				1142	2759	775	9.20	17.75	26.95	10000
				2585	5747	300	26.21	20.95	22.58	50000
VI.1	1000	5×10^5	522	1	393	0	6.73	2.62	9.35	5000
				1	367	23	13.95	23.90	37.85	50000
				1	424	1	6.05	121.59	127.64	250000
VI.2	1000	5×10^5	560	1	371	2320	17.16	5.72	22.88	5000
				1	313	13	8.66	16.75	25.41	50000
				1	308	16	5.81	117.55	123.36	250000
VI.3	1000	5×10^5	559	1	339	21	5.02	3.46	8.48	5000
				1	258	1488	3.50	13.86	17.36	50000
				1	289	559	3.13	106.23	109.36	250000
VII.3	1250	10^5	250	59973	123071	175	3.90	119.2	123.1	1000
				116838	237785	268	3.23	218.62	221.85	10000
				177232	359508	165	2.13	342.79	344.92	50000
VIII.1	1250	5×10^5	557	189	823	67	8.12	3.72	11.84	5000
				477	1399	145	10.98	21.32	32.30	50000
				306	1195	76	6.46	101.54	108.00	250000
VIII.2	1250	5×10^5	555	71	549	71	16.54	5.27	21.81	5000
				38	505	78	9.02	20.65	29.67	50000
				1	577	4	8.79	109	117.79	250000
VIII.3	1250	5×10^5	558	107	1023	86	23.46	19.31	42.77	5000
				159	1891	50	14.47	29.35	43.82	50000
				107	833	43	5.49	102.69	108.18	250000

Table 2: Computational results for `Exact-SIBLP` using different J_0 set and $m \in \{1000, 1250\}$

gap in parenthesis.

Based on the results in Tables 1 and 2, we observe that the size of initial set J_0 impacts the CG-Time and BnB-Time. With the increase in $|J_0|$, BnB-Time increases and CG-Time decreases for most of the instances. This impacts T-Time. For 63% of the instances, selecting $|J_0|$ as 1% of the total number of constraints, i.e., $|\mathcal{S}|$, leads to least total solution time. Additionally, as per the results in Table 3, the current implementation of `Exact-SIBLP` outperforms Gurobi and `Exact-IHSP`. Specifically, total time taken by `Exact-SIBLP`, which includes CG-Time, is on average 13.41% of the time taken by Gurobi. By excluding CG-Time, this percentage reduces to 9.63%. As expected, `Exact-SIBLP` explores 181% of the number of nodes explored by Gurobi with default settings. Note that the '0' entry in the #Nodes column implies that Gurobi solved the instance using its presolve heuristics. For the instances solved by `Exact-SIBLP` within the time limit, the total time taken by `Exact-SIBLP` is on average 3.23% of the time taken by `Exact-IHSP`.

5 Conclusion

We studied semi-infinite disjunctive programs (SIDPs), semi-infinite mixed binary linear programs (SIMBLPs), and semi-infinite binary programs with implicitly defined constraints (SIBLP-IC). We presented (*i*) a linear programming equivalent for SIDPs, (*ii*) a hierarchy of relaxations, sequential convexification approach and partial convex hulls for SIMBLPs, and (*iii*) valid inequalities for SIDP and SIMBLP, and (*iv*) an exact algorithm for solving SIBLP-IC. These results also established a direct connection between reformulation linearlization technique for finite and semi-infinite mixed binary programs and disjunctive programming that does not require the computation of projections multiple times. Additionally, we demonstrated the applications of our approaches for solving implicit hitting set problem, disjunctive convex programs, SIMBLPs with discrete variables, robust 0-1 knapsack problems, distributionally robust, and distributionally robust chance constrained problems. We evaluated performance of the exact algorithm for SIBLP-IC by introducing set-covering problem with infinite demand points and observed that this approach is computationally efficient.

Acknowledgment. This research is funded by National Science Foundation Grant CMMI 1824897, which is gratefully acknowledged.

References

Egon Balas. Disjunctive Programming. In P.L. Hammer, E. L. Johnson, and B. H. Korte, editors, *Annals of Discrete Mathematics*, volume 5 of *Discrete Optimization II Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium*, pages 3–51. Elsevier, 1979.

Egon Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1–3):3–44, December 1998.

Inst#	m	$\lvert\mathcal{S}\rvert$	Exact-SIBLP							Gurobi			Exact-IHSP	
			#Nodes	#CG-O	#Cuts	CG-Time	BnB-Time	T-Time	OPT	Time	#Nodes	OPT	Time	OPT
I.1	500	5×10^4	1	336	6	0.92	0.8	1.72	181	10.18	1	181	31.27	181
I.2	500	5×10^4	131	630	28	1.13	1.02	2.15	178	10.23	1	178	31.54	178
I.3	500	5×10^4	11	473	0	1.46	1.08	2.54	183	10.25	1	183	32.71	183
II.1	500	10^5	1	266	12	1.59	0.30	1.89	259	27.27	1	259	79.15	259
II.2	500	10^5	1	304	5	1.79	0.73	2.52	267	25.23	1	259	89.11	267
II.3	500	10^5	159	647	72	1.75	1.32	3.07	224	28.31	1	224	89.11	224
III.1	750	10^5	71	506	28	2.38	2.25	4.63	258	43.35	1	258	109.63	258
III.2	750	10^5	71	542	44	3.55	1.56	5.11	250	53.07	209	250	150.94	250
III.3	750	10^5	390	1147	39	2.33	2.59	4.92	258	47.97	262	258	122.85	258
IV.1	750	5×10^5	1	293	9	4.57	1.09	5.66	538	270.64	0	538	594.20	538
IV.2	750	5×10^5	1	258	2142	2.66	0.71	3.37	530	171	0	530	502.75	530
IV.3	750	5×10^5	1	289	14	2.27	1.09	3.36	536	151.8	0	536	519.1	536
V.1	1000	10^5	3993	8558	0	1.04	15.25	16.29	251	48.2	4542	251	351.35	251
V.2	1000	10^5	23665	49007	100	5.09	48.62	53.71	237	127.1	25837	237	542.81	237
V.3	1000	10^5	2585	5714	300	1.69	20.89	22.58	255	64.88	1596	255	294.57	255
VI.1	1000	5×10^5	1	393	0	6.73	2.62	9.35	522	353.3	1	522	1257	522
VI.2	1000	5×10^5	1	371	2320	17.16	5.72	22.88	560	359.9	1	560	1254	560
VI.3	1000	5×10^5	1	339	21	5.02	3.46	8.48	559	343.5	1	559	873.8	559
VII.1	1250	10^5	1.8×10^6	3.8×10^6	204	13.47	3600+	3600+	228 (0.9%)	3600+	9×10^5	228 (0.9%)	3600+	237 (4.2%)
VII.2	1250	10^5	1.5×10^6	3.2×10^6	161	10.86	3600+	3600+	242 (0.4%)	3600+	9.3×10^5	242 (0.7%)	3600+	243 (0.4%)
VII.3	1250	10^5	59973	123071	175	3.90	119.20	123.10	250	255.3	86909	250	1847	250
VIII.1	1250	5×10^5	189	823	67	8.12	3.72	11.84	557	288	4385	557	1421	557
VIII.2	1250	5×10^5	71	549	71	16.54	5.27	21.81	555	433	1	555	1463	555
VIII.3	1250	5×10^5	107	1023	86	23.46	19.31	42.77	558	411.35	86	558	2167	558

Table 3: Computational Results for Set-Covering Problem Instances

Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming*, 58(1-3):295–324, January 1993.

Manish Bansal and Kiavash Kianfar. Planar maximum coverage location problem with partial coverage and rectangular demand and service zones. *INFORMS Journal on Computing*, 29(1): 152–169, 2017.

Ahron Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 2001. ISBN 978-0-89871-882-9.

A. Charnes, W. W. Cooper, and K. Kortanek. Duality in Semi-Infinite Programs and Some Works of Haar and Carathéodory. *Management Science*, 9(2):209–228, January 1963.

A. Charnes, W. W. Cooper, and K. Kortanek. On Representations of Semi-Infinite Programs which Have No Duality Gaps. *Management Science*, 12(1):113–121, September 1965.

John W. Chinneck. Fast Heuristics for the Maximum Feasible Subsystem Problem. *INFORMS Journal on Computing*, 13(3):210–223, August 2001.

John W. Chinneck. The maximum feasible subset problem (maxFS) and applications. *INFOR: Information Systems and Operational Research*, 57(4):496–516, October 2019. ISSN 0315-5986.

Michel Gendreau, Gilbert Laporte, and Frédéric Semet. The Covering Tour Problem. *Operations Research*, 45(4):568–576, August 1997.

Miguel Angel Goberna and Marco Lopez-Cerda. *Linear semi-infinite optimization*. John Wiley & Sons Chichester, January 1998. ISBN 978-1-4899-8043-4. doi: 10.1007/978-1-4899-8044-1_3.

P. R. Gribik. A central-cutting-plane algorithm for semi-infinite programming problems. In R. Hettich, editor, *Semi-Infinite Programming*, Lecture Notes in Control and Information Sciences, pages 66–82, Berlin, Heidelberg, 1979. Springer.

S. A. Gustafson and K. O. Kortanek. Numerical treatment of a class of semi-infinite programming problems. *Naval Research Logistics Quarterly*, 20(3):477–504, 1973.

R. Hettich and K. O. Kortanek. Semi-Infinite Programming: Theory, Methods, and Applications. *SIAM Review*, 35(3):380–429, 1993.

Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations at the IBM Thomas J. Watson Research Center*, The IBM Research Symposia Series. Springer US, Boston, MA, 1972.

Christos Koufogiannakis and Neal E. Young. Beating Simplex for Fractional Packing and Covering Linear Programs. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 494–504, October 2007.

Erick Moreno-Centeno and Richard M. Karp. The Implicit Hitting Set Approach to Solve Combinatorial Optimization Problems with an Application to Multigenome Alignment. *Operations Research*, 61(2):453–468, April 2013.

Alan T. Murray. Geography in Coverage Modeling: Exploiting Spatial Structure to Address Complementary Partial Service of Areas. *Annals of the Association of American Geographers*, 95(4): 761–772, December 2005.

Alan T. Murray. Advances in location modeling: GIS linkages and contributions. *Journal of Geographical Systems*, 12(3):335–354, September 2010.

George Nemhauser and Laurence Wolsey. In *Integer and Combinatorial Optimization*, pages 1–766. John Wiley & Sons, Ltd, 1988.

Serge A. Plotkin, David B. Shmoys, and Eva Tardos. Fast Approximation Algorithms for Fractional Packing and Covering Problems. *Mathematics of Operations Research*, 20(2):257–301, 1995.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970. ISBN 978-0-691-01586-6.

H. Sherali and W. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3): 411–430, August 1990.

Hanif D. Sherali and Warren P. Adams. A Reformulation-Linearization Technique (RLT) for semi-infinite and convex programs under mixed 0-1 and general discrete restrictions. *Discrete Applied Mathematics*, 157(6):1319–1333, March 2009.