A Finitely Convergent Disjunctive Cutting Plane Algorithm for Bilinear Programming

Hamed Rahimian*[1] and Sanjay Mehrotra†[2]

[1]Department of Industrial Engineering, Clemson University, Clemson SC 29634, USA
[2]Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston IL 60208, USA

Abstract

In this paper, we present and analyze a finitely-convergent disjunctive cutting plane algorithm to obtain an ϵ-optimal solution or detect the infeasibility of a general nonconvex continuous bilinear program. While the cutting planes are obtained like Saxena, Bonami, and Lee [Math. Prog. 130: 359–413, 2011] and Fampa and Lee [J. Global Optim. 80: 287–305, 2021], a feature of the algorithm that guarantees finite convergence is exploring near-optimal extreme point solutions to a current relaxation at each iteration. In this sense, the presented algorithm and its analysis extend the work Owen and Mehrotra [Math. Prog. 89: 437–448, 2001] for solving mixed-integer linear programs to the general bilinear programs.

Key words: Bilinear programming, Nonconvex programming, Disjunctive programming, Global optimization, Cutting planes

1 Introduction

In this paper, we study a general nonconvex continuous bilinear program (BLP) defined as follows:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y}} \ & \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{x}^\top \boldsymbol{A}_0 \boldsymbol{y} \\ \text{s.t.} \ & \boldsymbol{f}_\iota^\top \boldsymbol{x} + \boldsymbol{g}_\iota^\top \boldsymbol{y} + \boldsymbol{x}^\top \boldsymbol{A}_\iota \boldsymbol{y} + b_\iota \leq 0, \quad \iota \in [p] \\ & \underline{\boldsymbol{x}} \leq \boldsymbol{x} \leq \overline{\boldsymbol{x}},\ \underline{\boldsymbol{y}} \leq \boldsymbol{y} \leq \overline{\boldsymbol{y}}, \end{aligned} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{A}_\iota$, $\iota \in 0 \cup [p]$, are $n \times m$ matrices, $\boldsymbol{f}_\iota \in \mathbb{R}^n$, $\boldsymbol{g}_\iota \in \mathbb{R}^m$, $\iota \in 0 \cup [p]$, $\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}} \in \mathbb{R}^n$, $\underline{\boldsymbol{y}}, \overline{\boldsymbol{y}} \in \mathbb{R}^m$, and $b_\iota \in \mathbb{R}$, $\iota \in [p]$. We do not consider any structure on the matrices $\boldsymbol{A}_\iota$, $\iota \in 0 \cup [p]$. A

*hrahimi@clemson.edu
†mehrotra@northwestern.edu

bilinear program of the form (1) finds various applications in production, location-allocation, and product distribution situations [1], pooling [19], trim-loss and cutting stock [13, 25], packing [16], network interdiction [7], and economic equilibrium [17].

The problem of generating relaxations of a bilinear program has been investigated in the literature. A common method to obtain a linear programming relaxation of a bilinear function xy is introducing a new variable w and then relaxing the constraint $w = xy$. When variables x and y are restricted to a box, McCormick [18] constructs a polyhedral relaxation for the bilinear set defined by $w = xy$. Al-Khayyal and Falk [2] show that this relaxation describes the convex hull of the bilinear set.

Discretizing a subset of continuous variables gives a mixed-integer BLP that approximates the original BLP, see, e.g., [22, 10]. Gupte et al. [12] obtain a mixed-integer linear programming reformulation of a mixed-integer BLP using the binary expansion of integer variables. By studying the polyhedral structure of the set arising from McCormick envelopes for an individual bilinear term, Gupte et al. [12] obtain the convex hull of these reformulated individual bilinear sets and use them in a branch-and-bound algorithm to solve the reformulated mixed-integer linear program.

Other relaxations based on the reformulation-linearization technique (RLT) [31], second-order cone programming (SOCP), see, e.g., [8, 28], and semidefinite programming (SDP), see, e.g., [4], have been applied to continuous BLPs. For the case that there is no interaction between the continuous variables $\boldsymbol{x}$ and $\boldsymbol{y}$, except for in the bilinear objective function, Sherali and Alameddine [32] develop a RLT-based relaxation that theoretically dominates the McCormick relaxation. Using this RLT-based relaxation, they propose a finitely-convergent branch-and-bound algorithm. Dey et al. [8] study a bilinear program of the form (1), where the variables can be partitioned into two sets such that fixing the variables in any of the sets results in a linear program. They show that the convex hull of the set induced by a single constraint is SOC representable in the extended space (see also [28] for results on a more general quadratic equation). The intersection of such sets gives a relaxation that is stronger than the standard SDP relaxation intersected with the boolean quadratic polytope [8].

In this paper, we focus on using the lift-and-project methodology and disjunctive programming [3]. Our motivation to use this framework is that it simultaneously takes into account convex and nonconvex constraints, see, e.g., [7, 29, 30]. An infinitely-convergent disjunctive sequential convexification procedure for a continuous bilinear set is studied in a companion paper [23].

Treating bilinear terms in the context of global optimization has also been studied in the literature [14, 36, 33, 9]. Konno [14] proposes an infinitely-convergent cutting plane procedure to obtain a solution differing in objective value from the global optimal value of the studied BLP by no more than a predetermined quantity $\epsilon > 0$. Vaish and Shetty [36] propose an infinitely-convergent cutting plane procedure to obtain a global optimal solution to the studied BLP. They also propose a finitely-convergent cutting plane algorithm to obtain a solution differing in objective value from the global optimal value by no more than ϵ. Sherali and Shetty [33] propose a finitely-convergent cutting plane algorithm to obtain a globally optimal solution by generating polar cuts at an extreme point solution and generating disjunctive cuts at other points. In the studied BLP in [14, 36, 33] it is assumed that variables $\boldsymbol{x}$ and $\boldsymbol{y}$ belong to their own polytopes and there is no nonlinearity in the constraints. Hence, the objective function is nonconvex, while the feasible region is convex.

For mixed-integer quadratically constrained quadratic programs, Saxena et al. [29] propose to obtain valid disjunctive cuts using the eigenvalue decomposition of the quadratic violation matrix. For a continuous BLP, with bilinear terms in the objective function and constraints, Fampa and Lee [9] further extend the approach in [29] using the singular value decomposition of the bilinear violation matrix (we shall shortly review this approach in Section 2). They conduct extensive computational experiments to assess the performance of this approach and methods that convert a bilinear program to a quadratic program with a symmetric matrix. For a general class of optimization problems, [6] study general-purpose cuts that account for global nonconvexity, and they show the finite convergence of their proposed pure cut-generation procedure, obtained based on intersection cuts from convex forbidden zones. They further develop this method for polynomial optimization (also applicable to BLPs), by deriving several families of maximal outer-product-free sets from 2×2 submatrices.

Although Fampa and Lee [9] are concerned with the global optimization of the studied BLP, they do not provide any theoretical result to guarantee that an optimal solution is found in a finite number of iterations. To close this gap, in this paper, a modification to the approach in [29, 9] is analyzed to guarantee a finitely-convergent disjunctive programming-based pure cutting plane approach. This modification is inspired by the cutting plane approach proposed in Owen and Mehrotra [21] in the context of solving mixed-integer linear programs with general integer variables. As in [21], a fundamental feature of the analyzed algorithm in this paper is to generate valid inequalities at *all* near-optimal extreme point solutions of the current polyhedral relaxation (near-optimal solutions are defined precisely in Section 4 as γ-optimal solutions with $\gamma > 0$). We theoretically analyze that modifying the idea investigated in Fampa and Lee [9] with this vertex exploration guarantees finite convergence. As mentioned earlier, another related work to ours is [6]. However, unlike [6], whose cut-generation procedure should be applied globally to all extreme points of a current relaxation, we use vertex exploration on only γ-optimal extreme point solutions to achieve finite convergence, with $\gamma > 0$. This results in an additional subtlety in the analysis as we will establish in Section 4. Additionally, from a computational perspective, for an appropriate choice of γ, the number of γ-optimal extreme points can be significantly smaller than the number of extreme points in the entire polytope.

Although, in theory, this vertex exploration guarantees finite convergence, there are some computational limitations. On the one hand, exploring all near-optimal extreme point solutions is computationally expensive. On the other hand, not all generated cuts through this vertex exploration would necessarily have computational values. Thus, a judicious (problem-dependent) vertex exploration is necessary for practical implementations. We provide some indications of these limitations in our numerical results in Section 5, where we generate cuts at only *a few* extreme point solutions. We conduct numerical experiments to compare an *implementation* of the idea proposed in Fampa and Lee [9] (see Section 2.2 for more details on our implementation), a *practical implementation* of the algorithm analyzed in this paper (see Section 5 for more details), and the pure cutting plane algorithm proposed in [6]. While we conduct a comparative study, our primary aim in this paper is not to conclude the computational efficiency of one algorithm over another but to provide a fundamental modification of the algorithm proposed in Fampa and Lee [9] that guarantees finite convergence.

To the best of our knowledge, the analyzed algorithm in this paper is the first pure cutting plane approach that solves (1) to ϵ-optimality (to be defined precisely in Section 4) or detects infeasibility in a finite number of iterations and only through a local vertex exploration of γ-optimal solutions. We emphasize that the feasible region in (1) is nonconvex. This is different from the studied BLP in [14, 36, 33], where an optimal solution is attained at an extreme point $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, with $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ to be the extreme points of their corresponding polytopes, see, e.g., [14, Theorem 2.1].

This paper is organized as follows. In Section 2, we review the lift-and-project methodology of Saxena et al. [29] in the context of a BLP, and the basic ideas of disjunctive programming. We also illustrate our motivation to theoretically enhance the procedure studied by [9]. In Section 3, we present the cut generation component of our analyzed algorithm. In Section 4, we analyze a disjunctive cutting plane algorithm that finds an ϵ-optimal solution to (1) or detect infeasibility in a finite number of iterations. In Section 5, we demonstrate the optimality gap improvement gained from a practical implementation of the analyzed cutting plane algorithms in this paper. We end with conclusions in Section 6.

Notation and Definitions: For two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\boldsymbol{A} \bullet \boldsymbol{B} = Tr(\boldsymbol{A}^\top \boldsymbol{B})$ denotes the Frobenius inner product between matrices. We let $[d]$ denote the index set $\{1, \dots, d\}$. Throughout this paper, vectors are denoted by boldface lowercase letters and matrices are denoted by boldface uppercase letters. Sets are denoted by calligraphic or normal uppercase letters. All sets in this paper are subsets of a finite-dimensional Euclidean space $\mathbb{R}^d$, for some $d > 0$. Consider a set $\mathcal{B} \subseteq \mathbb{R}^d$. Let $\text{ext}(\mathcal{B})$, $\text{cl}(\mathcal{B})$, and $\text{conv}\,(\mathcal{B})$ denote the set of extreme points, closure, and convex hull of the set $\mathcal{B}$. Let $\text{Proj}_{\boldsymbol{x}}(\mathcal{B})$ denote the projection of $\mathcal{B}$ onto the $\boldsymbol{x}$-space. Let $\boldsymbol{e}_i$ be the i-th unit vector in $\mathbb{R}^d$. Consider two sets $\mathcal{B}^1, \mathcal{B}^2 \subseteq \mathbb{R}^d$. The Hausdorff distance between $\mathcal{B}^1$ and $\mathcal{B}^2$ is denoted by $d_H(\mathcal{B}^1, \mathcal{B}^2)$ and is defined as $d_H(\mathcal{B}^1, \mathcal{B}^2) := \max\{\sup_{\boldsymbol{b}^2 \in \mathcal{B}^2} \inf_{\boldsymbol{b}^1 \in \mathcal{B}^1} \|\boldsymbol{b}^1 - \boldsymbol{b}^2\|, \sup_{\boldsymbol{b}^1 \in \mathcal{B}^1} \inf_{\boldsymbol{b}^2 \in \mathcal{B}^2} \|\boldsymbol{b}^1 - \boldsymbol{b}^2\|\}$. A sequence of sets $\{\mathcal{B}^t\}$ is called a decreasing sequence of nested sets if $\mathcal{B}^{t+1} \subseteq \mathcal{B}^t$, $t \geq 0$. We say that a sequence of closed sets $\{\mathcal{B}^t\}$ of $\mathbb{R}^d$ converges to a closed set $\bar{\mathcal{B}} \subseteq \mathbb{R}^d$ in Hausdorff distance, and denote it by $\text{h-lim}_{t\to\infty} \mathcal{B}^t = \bar{\mathcal{B}}$, if $d_H(\mathcal{B}^t, \bar{\mathcal{B}}) \to 0$ as $t \to \infty$. According to [27, Lemma 1], it means that either $\bar{\mathcal{B}}$ and $\mathcal{B}^t$ are empty for all $t \geq \bar{t}$ or for any $\delta > 0$, there exists $\hat{t} > 0$ such that for all $t \geq \hat{t}$, we have $\inf_{\boldsymbol{b} \in \bar{\mathcal{B}}} \|\boldsymbol{b} - \boldsymbol{b}^t\| \leq \delta$ for all $\boldsymbol{b}^t \in \mathcal{B}^t$ and $\inf_{\boldsymbol{b}^t \in \mathcal{B}^t} \|\boldsymbol{b} - \boldsymbol{b}^t\| \leq \delta$ for all $\boldsymbol{b} \in \bar{\mathcal{B}}$. For a sequence of sets $\{\mathcal{B}^t\}$ of $\mathbb{R}^d$, $\boldsymbol{b} \in \limsup_{t\to\infty} \mathcal{B}^t = \cap_{\hat{t}=1}^{\infty} \text{cl}(\cup_{t=\hat{t}}^{\infty} \mathcal{B}^t)$ if for any $\hat{t} > 0$, there exists $t \geq \hat{t}$ such that $\boldsymbol{b} \in \mathcal{B}^t$. We also have $\boldsymbol{b} \in \liminf_{t\to\infty} \mathcal{B}^t = \cup_{\hat{t}=1}^{\infty} \text{cl}(\cap_{t=\hat{t}}^{\infty} \mathcal{B}^t)$ if there exists $\hat{t} > 0$ such that for all $t \geq \hat{t}$, we have $\boldsymbol{b} \in \mathcal{B}^t$. We say that a sequence of sets $\{\mathcal{B}^t\}$ of $\mathbb{R}^d$ converges to $\bar{\mathcal{B}} \subseteq \mathbb{R}^d$ in the sense of Kuratowski, and denote it by $\mathcal{B}^t \xrightarrow{\text{K}} \bar{\mathcal{B}}$ as t $\to \infty$, if $\limsup_{t\to\infty} \mathcal{B}^t = \liminf_{t\to\infty} \mathcal{B}^t = \bar{\mathcal{B}}$. The following lemmas establish some relationships between Hausdorff and Kuratowski convergences and will be used in the sequel.

Lemma 1. *O'Searcoid [20, Theorem 12.1.3] A decreasing sequence of nonempty, nested, closed sets of a compact metric space has a nonempty compact intersection.*

Lemma 2. *Salinetti and Wets [27, Proposition 2] Suppose that $\{\mathcal{B}^t\}$ is a decreasing sequence of nested closed sets of a finite-dimensional Euclidean space. Then, $\{\mathcal{B}^t\}$ converges to $\cap_{t=1}^{\infty}\mathcal{B}^t$ in the sense of Kuratowski, as $t \to \infty$.*

Lemma 3. *Salinetti and Wets [27, Corollary 3A] Suppose that $\{\mathcal{B}^t\}$ is a sequence of nonempty compact connected[1] sets of a finite-dimensional Euclidean space. Then, $\text{h-lim}_{t\to\infty}\mathcal{B}^t = \bar{\mathcal{B}}$ if and only if $\mathcal{B}^t \xrightarrow{K} \bar{\mathcal{B}}$ as $t \to \infty$, i.e., the Hausdorff convergence implies the Kuratowski convergence and vice versa, and the limits are equal.*

Throughout the paper, we use the Hausdorff convergence and the Kuratowski convergence interchangeably under assumptions of Lemma 3. Moreover, we often work with a decreasing sequence of nested closed sets. For reference, the definition of the Hausdorff distance between two sets $\mathcal{B}^2 \subseteq \mathcal{B}^1 \subseteq \mathbb{R}^d$ simplifies to $d_H(\mathcal{B}^1, \mathcal{B}^2) = \sup_{\boldsymbol{b}^1\in\mathcal{B}^1} \inf_{\boldsymbol{b}^2\in\mathcal{B}^2} \|\boldsymbol{b}^1 - \boldsymbol{b}^2\|$.

2 Lift-and-Project Methodology of Saxena et al. [29], Fampa and Lee [9]

By introducing additional variables $W_{ij} = x_i y_j$, $i \in [n]$, $j \in [m]$, problem (1) can be equivalently written as the following nonlinear program in the lifted space:

$$\begin{aligned} \min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\mathcal{K}} \quad & \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W} \\ \text{s.t.} \quad & \boldsymbol{W} = \boldsymbol{x}\boldsymbol{y}^\top, \end{aligned} \tag{BLP}$$

where

$$\mathcal{K} := \left\{ (\boldsymbol{x},\boldsymbol{y},\boldsymbol{W}) \,\middle|\, \begin{array}{l} \boldsymbol{f}_\iota^\top \boldsymbol{x} + \boldsymbol{g}_\iota^\top \boldsymbol{y} + \boldsymbol{A}_\iota \bullet \boldsymbol{W} + b_\iota \le 0, \ \iota \in [p], \\ \underline{\boldsymbol{x}} \le \boldsymbol{x} \le \overline{\boldsymbol{x}}, \ \underline{\boldsymbol{y}} \le \boldsymbol{y} \le \overline{\boldsymbol{y}} \end{array} \right\}. \tag{2}$$

Set

$$\mathcal{F} := \{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W}) \in \mathcal{K} \,|\, \boldsymbol{W} = \boldsymbol{x}\boldsymbol{y}^\top\} \tag{3}$$

is the feasible region of (BLP), and set $\mathcal{K}$ is the feasible region of a relaxation of (BLP). Note that all the constraints in $\mathcal{K}$ are linear in $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{W}$, and $\mathcal{K}$ is a convex set. On the other hand, $\boldsymbol{W} = \boldsymbol{x}\boldsymbol{y}^\top$ induces a nonconvex region.

In this paper, we are interested in disjunctive programming procedures in the space of $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})$. A disjunctive programming procedure to treat the bilinear terms is studied in Fampa and Lee [9] by applying McCormick convexification of $\boldsymbol{W} = \boldsymbol{x}\boldsymbol{y}^\top$ and extending the ideas in Saxena et al. [29]

[1] Set $\mathcal{B}$ is not connected if there are two disjoint open sets $\mathcal{U}$ and $\mathcal{V}$ such that $\mathcal{B} \subset \mathcal{U} \cup \mathcal{V}$, $\mathcal{B} \cap \mathcal{U} \neq \emptyset$, and $\mathcal{B} \cap \mathcal{V} \neq \emptyset$.

for symmetric convex quadratic terms to bilinear terms. Because the approach in [29, 9] forms a basis for our work, let us first recall their procedure.

For any $\boldsymbol{u} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^m$, any feasible solution to (BLP) satisfies

$$\boldsymbol{u}^\top \boldsymbol{W} \boldsymbol{v} = (\boldsymbol{u}^\top \boldsymbol{x})(\boldsymbol{v}^\top \boldsymbol{y}). \tag{4}$$

Because $(\boldsymbol{u}^\top \boldsymbol{x})(\boldsymbol{v}^\top \boldsymbol{y}) = \left(\frac{\boldsymbol{u}^\top \boldsymbol{x}+\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2 - \left(\frac{\boldsymbol{u}^\top \boldsymbol{x}-\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2$, (4) is equivalent to the following two inequalities

$$\boldsymbol{u}^\top \boldsymbol{W} \boldsymbol{v} - \left(\frac{\boldsymbol{u}^\top \boldsymbol{x}+\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2 + \left(\frac{\boldsymbol{u}^\top \boldsymbol{x}-\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2 \le 0, \tag{5}$$

$$-\boldsymbol{u}^\top \boldsymbol{W} \boldsymbol{v} + \left(\frac{\boldsymbol{u}^\top \boldsymbol{x}+\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2 - \left(\frac{\boldsymbol{u}^\top \boldsymbol{x}-\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2 \le 0. \tag{6}$$

Observe that the concave terms $-\left(\frac{\boldsymbol{u}^\top \boldsymbol{x}+\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2$ and $-\left(\frac{\boldsymbol{u}^\top \boldsymbol{x}-\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2$, in (5) and (6), respectively, result in a nonconvex region. A way to handle this nonconvexity is to approximate the concave terms with their secant inequalities and to utilize disjuncntive programming to derive valid disjunctive cuts for conv $(\mathcal{F})$ [29]. More precisely, constraints (5) and (6) give rise to the following disjunction, which is satisfied by any feasible solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W})$ to (BLP):

$$\vee_{r=1}^2 \vee_{s=1}^2 \tilde{\mathcal{S}}_{rs}(\boldsymbol{c}, \tilde{\mathcal{K}}, \boldsymbol{\beta}), \tag{7}$$

where $\tilde{\mathcal{K}}$ is a (bounded) convex relaxation of $\mathcal{F}$ (e.g., $\mathcal{K}$), $\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{v})$, and

$$\tilde{\mathcal{S}}_{rs}(\boldsymbol{c}, \tilde{\mathcal{K}}, \boldsymbol{\beta}) := \left\{ (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \tilde{\mathcal{K}} \;\middle|\; \begin{array}{l} \beta_{1,r} \le \frac{\boldsymbol{u}^\top \boldsymbol{x}+\boldsymbol{v}^\top \boldsymbol{y}}{2} \le \beta_{1,r+1},\ \beta_{2,s} \le \frac{\boldsymbol{u}^\top \boldsymbol{x}-\boldsymbol{v}^\top \boldsymbol{y}}{2} \le \beta_{2,s+1}, \\ \boldsymbol{u}^\top \boldsymbol{W} \boldsymbol{v} - \left(\frac{\boldsymbol{u}^\top \boldsymbol{x}+\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)(\beta_{1,r} + \beta_{1,r+1}) + \beta_{1,r}\beta_{1,r+1} \\ + \left(\frac{\boldsymbol{u}^\top \boldsymbol{x}-\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2 \le 0, \\ -\boldsymbol{u}^\top \boldsymbol{W} \boldsymbol{v} - \left(\frac{\boldsymbol{u}^\top \boldsymbol{x}-\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)(\beta_{2,s} + \beta_{2,s+1}) + \beta_{2,s}\beta_{2,s+1} \\ + \left(\frac{\boldsymbol{u}^\top \boldsymbol{x}+\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2 \le 0 \end{array} \right\}, \tag{8}$$

for $r, s = 1, 2$. Disjunction (7) is obtained by simultaneously splitting the range $[\beta_{1,1}, \beta_{1,3}]$ of function $\frac{\boldsymbol{u}^\top \boldsymbol{x}+\boldsymbol{v}^\top \boldsymbol{y}}{2}$ over $\tilde{\mathcal{K}}$ into two intervals $[\beta_{1,1}, \beta_{1,2}]$ and $[\beta_{1,2}, \beta_{1,3}]$, and by splitting the range $[\beta_{2,1}, \beta_{2,3}]$ of function $\frac{\boldsymbol{u}^\top \boldsymbol{x}-\boldsymbol{v}^\top \boldsymbol{y}}{2}$ over $\tilde{\mathcal{K}}$ into two intervals $[\beta_{2,1}, \beta_{2,2}]$ and $[\beta_{2,2}, \beta_{2,3}]$. Moreover, the disjunction simultaneously constructs secant inequalities of functions $-\left(\frac{\boldsymbol{u}^\top \boldsymbol{x}+\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2$ and $-\left(\frac{\boldsymbol{u}^\top \boldsymbol{x}-\boldsymbol{v}^\top \boldsymbol{y}}{2}\right)^2$ in each corresponding interval. The breakpoints $\{\beta_{1,1}, \beta_{1,2}, \beta_{1,3}\}$ might have overlaps. However, as long as these breakpoints are in the range of function $\frac{\boldsymbol{u}^\top \boldsymbol{x}+\boldsymbol{v}^\top \boldsymbol{y}}{2}$, a disjunction of the form (7) is valid. A similar situation might happen for the breakpoints $\{\beta_{2,1}, \beta_{2,2}, \beta_{2,3}\}$.

For the rest of the paper, we refer to the last two constraints in (8) as *secant-induced* inequalities. We also let the index k represent the (r, s)-pair, where $k \in \{1, 2, 3, 4\}$. Hence, hereafter, we denote

$\tilde{\mathcal{S}}_{rs}(\boldsymbol{c}, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ as $\tilde{\mathcal{S}}_k(\boldsymbol{c}, \tilde{\mathcal{K}}, \boldsymbol{\beta})$. Let $r(k)$ and $s(k)$ denote the r and s component of the index k. For the ease of exposition, for $k \in \{1,2,3,4\}$, $\beta_{1,k}$ and $\beta_{2,k}$ should be understood as $\beta_{1,r(k)}$ and $\beta_{2,s(k)}$, respectively. Similarly, $\beta_{1,k+1}$ and $\beta_{2,k+1}$ should be understood as $\beta_{1,r(k)+1}$ and $\beta_{2,s(k)+1}$, respectively. We also denote a (bounded) convex relaxation of $\mathcal{F}$ by $\tilde{\mathcal{K}}$ throughout the paper.

2.1 Disjunctive Programming

Given a solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ to the current relaxation $\tilde{\mathcal{K}}$, Fampa and Lee [9] analyze the singular value decomposition (SVD) of $\hat{\boldsymbol{W}} - \hat{\boldsymbol{x}}\hat{\boldsymbol{y}}^\top$ to find suitable vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, corresponding to a nonzero singular value σ, i.e., $\boldsymbol{u}^\top(\hat{\boldsymbol{W}} - \hat{\boldsymbol{x}}\hat{\boldsymbol{y}}^\top)\boldsymbol{v} = \sigma \neq 0$. The vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are used in order to form a disjunction of the form (7), and subsequently, to derive disjunctive cuts for $\operatorname{conv}(\mathcal{F})$ through a cut-generation linear program (CGLP). The CGLP used in [9] contains linearization of the constraints in (8), where the convex quadratic terms are replaced by their outer approximations. Moreover, $\tilde{\mathcal{K}}$ includes the constraints of McCormick convexification of $\boldsymbol{W} = \boldsymbol{x}\boldsymbol{y}^\top$, using the box constraints on $\boldsymbol{x}$ and $\boldsymbol{y}$, and all the previously added disjunctive cuts. Below, we present an abstract form of this CGLP to generate a valid inequality to cut off the current solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$.

Lemma 4. *Consider a bounded polyhedral relaxation $\tilde{\mathcal{K}}$ of $\mathcal{F}$, a fixed vector $\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{v})$, and a choice of breakpoints $\boldsymbol{\beta}$ for a (2×2)-way disjunction (7). Let $\{\boldsymbol{a}_{\iota k}^\top \boldsymbol{x} + \boldsymbol{b}_{\iota k}^\top \boldsymbol{y} + \boldsymbol{C}_{\iota k} \bullet \boldsymbol{W} \geq d_{\iota k},\ \iota \in [l]\}$ represent the set of constraints in $\tilde{\mathcal{S}}_k(\boldsymbol{c}, \tilde{\mathcal{K}}, \boldsymbol{\beta})$, after linearization of the quadratic terms, for $k = 1,2,3,4$. Then, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \operatorname{conv}\left(\vee_{k=1}^4 \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \,|\, \boldsymbol{a}_{\iota k}^\top \boldsymbol{x} + \boldsymbol{b}_{\iota k}^\top \boldsymbol{y} + \boldsymbol{C}_{\iota k} \bullet \boldsymbol{W} \geq d_{\iota k},\ \iota \in [l]\}\right)$ if the optimal value of the following CGLP is nonnegative*

$$\min\ \boldsymbol{\alpha}^\top \hat{\boldsymbol{x}} + \boldsymbol{\theta}^\top \hat{\boldsymbol{y}} + \boldsymbol{H} \bullet \hat{\boldsymbol{W}} - \rho \tag{9a}$$

$$s.t.\ \boldsymbol{A}_k^\top \boldsymbol{\pi}_k = \boldsymbol{\alpha},\ \forall k, \tag{9b}$$

$$\boldsymbol{B}_k^\top \boldsymbol{\pi}_k = \boldsymbol{\theta},\ \forall k, \tag{9c}$$

$$\boldsymbol{D}_{jk}^\top \boldsymbol{\pi}_k = \boldsymbol{h}_j,\ j \in [m], \forall k, \tag{9d}$$

$$\boldsymbol{d}_k^\top \boldsymbol{\pi}_k \geq \rho,\ \forall k, \tag{9e}$$

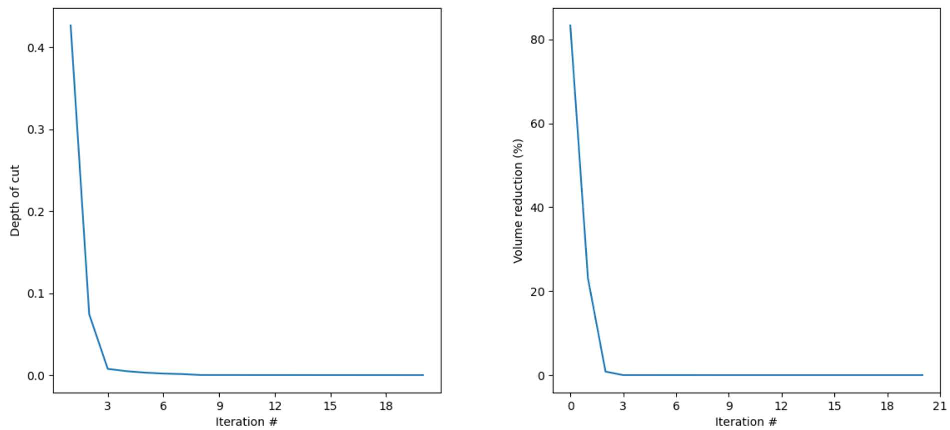
$$\boldsymbol{\pi}_k \geq \boldsymbol{0},\ \forall k, \tag{9f}$$

where row ι of $\boldsymbol{d}_k$, $\boldsymbol{A}_k$, and $\boldsymbol{B}_k$ is composed of $d_{\iota k}$, $\boldsymbol{a}_{\iota k}^\top$, and $\boldsymbol{b}_{\iota k}^\top$, respectively. Moreover, row ι of $\boldsymbol{D}_{jk}$ is column j of $\boldsymbol{C}_{\iota k}$, and column j of $\boldsymbol{H}$ is composed of $\boldsymbol{h}_j$, $j \in [m]$. If the optimal value of CGLP (9) is negative and $(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{H}, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_4)$ is an optimal solution to (9), then $\boldsymbol{\alpha}^\top \boldsymbol{x} + \boldsymbol{\theta}^\top \boldsymbol{y} + \boldsymbol{H} \bullet \boldsymbol{W} \geq \rho$ is a valid inequality for $\operatorname{conv}\left(\vee_{k=1}^4 \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \,|\, \boldsymbol{a}_{\iota k}^\top \boldsymbol{x} + \boldsymbol{b}_{\iota k}^\top \boldsymbol{y} + \boldsymbol{C}_{\iota k} \bullet \boldsymbol{W} \geq d_{\iota k},\ \iota \in [l]\}\right)$, which cuts off $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$.

Proof. The result follows from an application of Balas [3, Theorem 3.1]. □

2.2 Motivating Examples

It is illustrated in the numerical experiments of [9] that their proposed procedure is not guaranteed to reach an optimal solution of (BLP) or have a slow rate of convergence. In two simple examples,



(a) Depth of disjunctive cut.

(b) Percentage of the reduction in the volume of the current relaxation.

Figure 1: Depth of disjunctive cuts and percentage of the reduction in the volume of the current relaxation by `SVD` for Example 1.

we illustrate these issues.

Example 1. *Given* $\mathcal{K} = \left\{(x, y, W)\middle|\ x + 0.5y - 1 \leq 0,\ 0 \leq x \leq 1,\ 0 \leq y \leq 2\right\}$*, define problem* $\min_{(x,y,W)\in\mathcal{K}} x - y - 2W$*, where the optimal value is* -2.0625 *and the optimal solution is* $x^* = 0.125, y^* = 1.75$*. Consider an iterative algorithm that given a solution* $(\hat{x}, \hat{y}, \hat{W})$ *to the current relaxation* $\tilde{\mathcal{K}}$*, a disjunctive cut is generated if* $\hat{W} \neq \hat{x}\hat{y}$*. In order to form a* (2×2)*-way disjunction* (7)*, the breakpoints* $\beta_{1,2}$ *and* $\beta_{2,2}$ *are as follows:* $\beta_{1,2} = \frac{\hat{x}+\hat{y}}{2}$ *and* $\beta_{2,2} = \frac{\hat{x}-\hat{y}}{2}$*. The other breakpoints are obtained based on the lower and upper bounds of functions* $\frac{x+y}{2}$ *and* $\frac{x-y}{2}$ *over* $\tilde{\mathcal{K}}$*. A disjunctive cut is then obtained using the CGLP* (9)*, stated in Lemma 4, where we approximated the convex quadratic terms at 2500 equally-spaced points on* $[0, 1] \times [0, 2]$*. We refer to this particular implementation of the methodology described in [9], as "`SVD`" (standing for singular value decomposition). Note that for this example, left- and right-singular vectors are just the standard basis in* $\mathbb{R}$*. When `SVD` terminates after 20 iterations, we obtain a lower bound of* -2.0638*, with an optimality gap of* 0.0616%*. Figure 1a depicts the depth of each disjunctive cut, measured by the distance of the current solution to the induced disjunctive cut. Moreover, Figure 1b depicts the percentage of the reduction in the volume of the current relaxation. Observe from Figure 1a that `SVD` shows a very slow rate of convergence, where the disjunctive cuts have a depth of almost zero—and hence, cutting off only a negligible part of the polytope— while the solutions generated by the algorithm are converging to* $(0.145, 1.711)$*. We note that to calculate the volume of the current relaxation, we used a Python wrapper for Qhull library [5].*

In Section 5, we show that the proposed algorithm in this paper will attain a lower bound of 2.0629*, with an optimality gap of* 0.0234%*. Moreover, the solutions generated by the algorithm are converging to* $(0.128, 1.744)$*.*

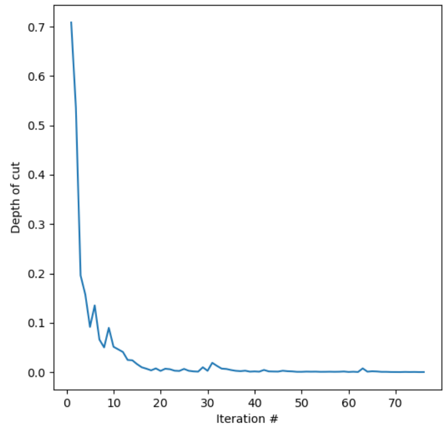


Figure 2: Depth of disjunctive cuts by `SVD` for Example 2.

Example 2. *Consider a problem of the form* (BLP)*, where* $\boldsymbol{x} \in \mathbb{R}^2$*,* $\boldsymbol{y} \in \mathbb{R}^2$*, and there is only one linear constraint connecting* $\boldsymbol{x}$*,* $\boldsymbol{y}$*, and* $\boldsymbol{W}$ *as follows:*

$$\underline{\boldsymbol{x}} = [0,0]^\top,\ \overline{\boldsymbol{x}} = [2,4]^\top,\ \boldsymbol{f}_0 = [1,2]^\top,\ \boldsymbol{f}_1 = [2,0.5]^\top,$$
$$\underline{\boldsymbol{y}} = [0,0]^\top,\ \overline{\boldsymbol{y}} = [1,2]^\top,\ \boldsymbol{g}_0 = [1,1]^\top,\ \boldsymbol{g}_1 = [2,1]^\top,$$
$$\boldsymbol{A}_0 = [-1,-2.5;-1,-3],\ \boldsymbol{A}_1 = [1,1;1,1],\ b = -3.$$

At each iteration, given a solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ *to the current relaxation* $\tilde{\mathcal{K}}$*, we obtain the left- and right-singular vectors* $\boldsymbol{u}$ *and* $\boldsymbol{v}$*, respectively, corresponding to the largest singular value of* $\hat{\boldsymbol{W}} - \hat{\boldsymbol{x}}\hat{\boldsymbol{y}}^\top$*. Then, using* $\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{v})$*, we form a* (2×2)*-way disjunction* (7) *and obtain a disjunctive cut using the CGLP* (9)*, stated in Lemma 4, where the convex quadratic terms are replaced by their outer approximations at the current solution. To form the disjunction, we choose the breakpoints* $\beta_{1,2}$ *and* $\beta_{2,2}$ *using the current solution as follows:* $\beta_{1,2} = \frac{\boldsymbol{u}^\top \hat{\boldsymbol{x}} + \boldsymbol{v}^\top \hat{\boldsymbol{y}}}{2}$ *and* $\beta_{2,2} = \frac{\boldsymbol{u}^\top \hat{\boldsymbol{x}} - \boldsymbol{v}^\top \hat{\boldsymbol{y}}}{2}$*. The other breakpoints are due to the lower and upper bounds of functions* $\frac{\boldsymbol{u}^\top \boldsymbol{x} + \boldsymbol{v}^\top \boldsymbol{y}}{2}$ *and* $\frac{\boldsymbol{u}^\top \boldsymbol{x} - \boldsymbol{v}^\top \boldsymbol{y}}{2}$ *over* $\tilde{\mathcal{K}}$*. When* `SVD` *terminates after 76 iterations, we obtain a lower bound* -0.5960*, while the optimal value to this problem is* -0.5*.* `SVD` *leaves an optimality gap of* 19.20%*. Figure 2 shows the depth of each disjunctive cut, indicating that* `SVD` *has a slow rate of convergence while trying to converge to the optimal solution* $\boldsymbol{x}^* = [0,1]^\top$*,* $\boldsymbol{y}^* = [0,1.25]^\top$*.*

Alternatively, consider a procedure that generates disjunctive cuts based on standard bases of $\mathbb{R}^2$*, and for all* $i \in \{1,2\}$*,* $j \in \{1,2\}$ *such that* $\hat{W}_{ij} \neq \hat{x}_i \hat{y}_j$*. Using* $(\boldsymbol{e}_i, \boldsymbol{e}_j)$*, we form a* (2×2)*-way disjunction, where the breakpoints* $\beta_{1,2}$ *and* $\beta_{2,2}$ *are as follows:* $\beta_{1,2} = \frac{\hat{x}_i + \hat{y}_j}{2}$ *and* $\beta_{2,2} = \frac{\hat{x}_i - \hat{y}_j}{2}$*. The other breakpoints are due to the lower and upper bounds of functions* $\frac{x_i + y_j}{2}$ *and* $\frac{x_i - y_j}{2}$ *over* $\tilde{\mathcal{K}}$*. We refer to this algorithm as "*`STD`*" (standing for the standard basis), and compare the results with* `SVD`*.* `STD` *yields a lower bound* -0.7068 *after 58 iterations and adding 211 cuts, leaving an optimality gap of* 41.36%*. As expected, algorithm* `STD` *yielded a worse lower bound than* `SVD`*, because* `STD` *is focused on only one bilinear term at a time, while* `SVD` *has a holistic view of all the bilinear terms.*

We emphasize that the `SVD` algorithm implemented to obtain the results in Examples 1 and 2 is not an exact reproduction of [9] for at least one reason. The authors in [9] choose uniformly-spaced breakpoints, e.g., $\beta_{1,2} = \frac{\beta_{1,1}+\beta_{1,3}}{2}$. On the contrary, for Examples 1 and 2, and all subsequent experiments in Section 5, we use a specific "solution-dependent" construction to choose the breakpoints (see more details in Section 3.2), inspired by the construction in [29] for quadratically-constrained quadratic programs.

Nevertheless, inspired by Examples 1 and 2, in Section 4, we analyze a finitely-convergent algorithm to reach an ϵ-optimal solution or detect infeasibility of (BLP) using any finite collection of bases $(\boldsymbol{u}, \boldsymbol{v})$ for $\mathbb{R}^n$ and $\mathbb{R}^m$, including the standard bases. We then extend the analyzed algorithm to the case that the bases are found through the SVD of the residual matrix $\boldsymbol{W} - \boldsymbol{x}\boldsymbol{y}^\top$, where a finite collection of bases is generated sequentially. We emphasize that our goal in this paper is not to propose an algorithm that is necessarily superior to the algorithm investigated in [9] in terms of the computational time—partially due to the reasons laid out above—but to analyze a fundamental modification to that algorithm that guarantees finite convergence.

3 Separating Inequalities and Minimum Distance Problem

A key observation around which this paper is developed is a reformulation of (BLP), discussed in Section 3.1. Then, we discuss the cut-generation component of the analyzed finitely-convergent algorithms. In Section 3.2, we present valid disjunctions. In Section 3.3, we describe valid separating inequalities and the corresponding projection problem.

3.1 Problem Reformulation

As explained in (4), we have

$$\mathcal{F} := \left\{ (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \mathcal{K} \;\middle|\; \begin{array}{l} \boldsymbol{u}^\top \boldsymbol{W} \boldsymbol{v} = (\boldsymbol{u}^\top \boldsymbol{x})(\boldsymbol{v}^\top \boldsymbol{y}), \\ \forall \boldsymbol{u} \in \mathbb{R}^n, \boldsymbol{v} \in \mathbb{R}^m, \text{ with } \|\boldsymbol{u}\| = 1, \; \|\boldsymbol{v}\| = 1 \end{array} \right\}. \tag{10}$$

A set closely related set is

$$\bar{\mathcal{F}}^\epsilon := \left\{ (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \mathcal{K} \;\middle|\; \begin{array}{l} |\boldsymbol{u}^\top \boldsymbol{W} \boldsymbol{v} - (\boldsymbol{u}^\top \boldsymbol{x})(\boldsymbol{v}^\top \boldsymbol{y})| \leq mn\epsilon, \\ \forall \boldsymbol{u} \in \mathbb{R}^n, \boldsymbol{v} \in \mathbb{R}^m, \text{ with } \|\boldsymbol{u}\| = 1, \; \|\boldsymbol{v}\| = 1 \end{array} \right\}, \tag{11}$$

for $\epsilon > 0$. A key observation to analyze the algorithms in this paper is that $\mathcal{F}$ can be equivalently reformulated with a *finite* number of nonlinear constraints, corresponding to the bases of $\mathbb{R}^n$ and $\mathbb{R}^m$. We will also show that a conservative approximation to $\bar{\mathcal{F}}^\epsilon$ can be reformulated with a finite number of nonlinear constraints. Observe that if $\bar{\mathcal{F}}^\epsilon$ is an empty set, then $\mathcal{F}$ is an empty set as well. Also, note that for both sets $\mathcal{F}$ and $\bar{\mathcal{F}}^\epsilon$, we restrict $\boldsymbol{u}$ and $\boldsymbol{v}$ to be unit vectors, without loss of generality. In the remainder of the paper, we may implicitly drop these restrictions from the set definition to simplify the exposition.

Proposition 1. *Let* $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$ *denote a set of mutually orthonormal vectors in* $\mathbb{R}^n$*, and* $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ *denote a set of mutually orthonormal vectors in* $\mathbb{R}^m$*. Then,* (BLP) *can be equivalently written as*

$$\begin{aligned} \min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\mathcal{K}} \quad & \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W} \\ \textit{s.t.} \quad & \boldsymbol{u}_i^\top \boldsymbol{W} \boldsymbol{v}_j = (\boldsymbol{u}_i^\top \boldsymbol{x})(\boldsymbol{v}_j^\top \boldsymbol{y}),\ i \in [n],\ j \in [m]. \end{aligned} \tag{BLP}$$

Proof. Observe that $\boldsymbol{W} = \boldsymbol{x}\boldsymbol{y}^\top \Rightarrow \boldsymbol{u}^\top \boldsymbol{W} \boldsymbol{v} = (\boldsymbol{u}^\top \boldsymbol{x})(\boldsymbol{v}^\top \boldsymbol{y})\ \forall \boldsymbol{u} \in \mathbb{R}^n, \boldsymbol{v} \in \mathbb{R}^m$, including $\boldsymbol{u}_i$, $\boldsymbol{v}_j$, $i \in [n]$, $j \in [m]$. We show that if $\boldsymbol{u}_i^\top \boldsymbol{W} \boldsymbol{v}_j = (\boldsymbol{u}_i^\top \boldsymbol{x})(\boldsymbol{v}_j^\top \boldsymbol{y}),\ \forall i \in [n],\ j \in [m] \Rightarrow \boldsymbol{W} = \boldsymbol{x}\boldsymbol{y}^\top$. Because $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$ is orthonormal, any $\boldsymbol{u} \in \mathbb{R}^n$ can be written as $\boldsymbol{u} = \sum_{i=1}^n \lambda_i \boldsymbol{u}_i$ for some $\boldsymbol{\lambda} \in \mathbb{R}^n$. Similarly, any $\boldsymbol{v} \in \mathbb{R}^m$ can be written as $\boldsymbol{v} = \sum_{j=1}^m \mu_j \boldsymbol{v}_j$ for some $\boldsymbol{\mu} \in \mathbb{R}^m$. Thus, we have

$$\begin{aligned} & \boldsymbol{u}_i^\top \boldsymbol{W} \boldsymbol{v}_j = (\boldsymbol{u}_i^\top \boldsymbol{x})(\boldsymbol{v}_j^\top \boldsymbol{y}),\ \forall i \in [n],\ j \in [m] \\ \Rightarrow & \left(\sum_{i=1}^n \lambda_i \boldsymbol{u}_i^\top\right) \boldsymbol{W} \boldsymbol{v}_j = \left(\sum_{i=1}^n \lambda_i \boldsymbol{u}_i^\top \boldsymbol{x}\right) \left(\boldsymbol{v}_j^\top \boldsymbol{y}\right),\ \forall \boldsymbol{\lambda} \in \mathbb{R}^n,\ j \in [m], \\ \Rightarrow & \left(\sum_{i=1}^n \lambda_i \boldsymbol{u}_i^\top\right) \boldsymbol{W} \left(\sum_{i=1}^n \mu_j \boldsymbol{v}_j\right) = \left(\sum_{i=1}^n \lambda_i \boldsymbol{u}_i^\top \boldsymbol{x}\right) \left(\sum_{j=1}^m \mu_j \boldsymbol{v}_j^\top \boldsymbol{y}\right),\ \forall \boldsymbol{\lambda} \in \mathbb{R}^n,\ \boldsymbol{\mu} \in \mathbb{R}^m \\ \Rightarrow & \boldsymbol{u}^\top \boldsymbol{W} \boldsymbol{v} = (\boldsymbol{u}^\top \boldsymbol{x})(\boldsymbol{v}^\top \boldsymbol{y}),\ \forall \boldsymbol{u} \in \mathbb{R}^n, \boldsymbol{v} \in \mathbb{R}^m \\ \Rightarrow & \boldsymbol{W} = \boldsymbol{x}\boldsymbol{y}^\top, \end{aligned}$$

by taking $\boldsymbol{u}$ and $\boldsymbol{v}$ be the bases vectors. Consequently, the result follows. □

Proposition 2. *Let* $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$ *denote a set of mutually orthonormal vectors in* $\mathbb{R}^n$*, and* $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ *denote a set of mutually orthonormal vectors in* $\mathbb{R}^m$*. Then, if* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \mathcal{F}^\epsilon$*, then* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \bar{\mathcal{F}}^\epsilon$*, where*

$$\mathcal{F}^\epsilon := \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \mathcal{K} \,|\, |\boldsymbol{u}_i^\top \boldsymbol{W} \boldsymbol{v}_j - (\boldsymbol{u}_i^\top \boldsymbol{x})(\boldsymbol{v}_j^\top \boldsymbol{y})| \le \epsilon,\ i \in [n], j \in [m]\}. \tag{12}$$

Proof. Consider unit vectors $\boldsymbol{u} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^m$. Let us write $\boldsymbol{u} = \sum_{i=1}^n \lambda_i \boldsymbol{u}_i$ for some $\boldsymbol{\lambda} \in \mathbb{R}^n$, and $\boldsymbol{v} = \sum_{j=1}^m \mu_j \boldsymbol{v}_j$ for some $\boldsymbol{\mu} \in \mathbb{R}^m$. First, note that we have $\boldsymbol{u}^\top \boldsymbol{u} = \boldsymbol{\lambda}^\top \boldsymbol{U}^\top \boldsymbol{U} \boldsymbol{\lambda}$, where $\boldsymbol{U}$ is a matrix whose columns are $\boldsymbol{u}_i$, $i = 1, \ldots, n$. Because $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{u}^\top \boldsymbol{u} = 1$, we have $\boldsymbol{\lambda}^\top \boldsymbol{\lambda} = 1$. Thus, $|\lambda_i| \le 1$ for $i \in [n]$. Similarly, we have $|\mu_j| \le 1$ for $j \in [m]$. Observe that

$$\begin{aligned} \boldsymbol{u}^\top \boldsymbol{W} \boldsymbol{v} - (\boldsymbol{u}^\top \boldsymbol{x})(\boldsymbol{v}^\top \boldsymbol{y}) =& \boldsymbol{u}^\top (\boldsymbol{W} - \boldsymbol{x}\boldsymbol{y}^\top) \boldsymbol{v} \\ =& \left(\sum_{i=1}^n \lambda_i \boldsymbol{u}_i\right)^\top (\boldsymbol{W} - \boldsymbol{x}\boldsymbol{y}^\top) \left(\sum_{j=1}^m \mu_j \boldsymbol{v}_j\right) \\ =& \sum_{i=1}^n \sum_{j=1}^m \lambda_i \mu_j \boldsymbol{u}_i^\top (\boldsymbol{W} - \boldsymbol{x}\boldsymbol{y}^\top) \boldsymbol{v}_j. \end{aligned}$$

Thus, for $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \mathcal{F}^\epsilon$, we have

$$\begin{aligned}
\left|\boldsymbol{u}^\top \boldsymbol{W}\boldsymbol{v} - (\boldsymbol{u}^\top \boldsymbol{x})(\boldsymbol{v}^\top \boldsymbol{y})\right| \leq & \left|\sum_{i=1}^n \sum_{j=1}^m \lambda_i \mu_j \boldsymbol{u}_i^\top (\boldsymbol{W} - \boldsymbol{x}\boldsymbol{y}^\top)\boldsymbol{v}_j\right| \\
\leq & \sum_{i=1}^n \sum_{j=1}^m \left|\lambda_i \mu_j \boldsymbol{u}_i^\top (\boldsymbol{W} - \boldsymbol{x}\boldsymbol{y}^\top)\boldsymbol{v}_j\right| \\
\leq & \max_{i=1}^n \max_{j=1}^m |\lambda_i \mu_j| \times \sum_{i=1}^n \sum_{j=1}^m \left|\boldsymbol{u}_i^\top (\boldsymbol{W} - \boldsymbol{x}\boldsymbol{y}^\top)\boldsymbol{v}_j\right| \\
\leq & \sum_{i=1}^n \sum_{j=1}^m \left|\boldsymbol{u}_i^\top (\boldsymbol{W} - \boldsymbol{x}\boldsymbol{y}^\top)\boldsymbol{v}_j\right| \\
\leq & mn\epsilon.
\end{aligned}$$

This completes the proof. □

Throughout this section, we assume that $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$ is a set of mutually orthonormal vectors in $\mathbb{R}^n$, and $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ is a set of mutually orthonormal vectors in $\mathbb{R}^m$. For ease of exposition, let the index a represent the (i,j)-pair, where $a \in [nm]$. For the index a, we denote $(\boldsymbol{u}_a, \boldsymbol{v}_a)$ by $\boldsymbol{c}_a$.

3.2 Valid Single-Vector Disjunction

Consider a (bounded) convex relaxation $\tilde{\mathcal{K}}$ of $\mathcal{F}$ and $\boldsymbol{c}_a$, $a \in [nm]$. Let us define the following set:

$$\mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta}) := \operatorname{conv}\left(\vee_{k=1}^4 \tilde{\mathcal{S}}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})\right). \tag{13}$$

A cutting plane is generated based on a single-vector disjunction. Given a relaxation $\tilde{\mathcal{K}}$ of $\mathcal{F}$, let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ be an optimal extreme point solution to the minimization problem over $\tilde{\mathcal{K}}$ that needs to be cut off by a valid linear inequality. In particular, suppose that the solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ is not satisfying the constraint $\boldsymbol{u}_a^\top \boldsymbol{W}\boldsymbol{v}_a = (\boldsymbol{u}_a^\top \boldsymbol{x})(\boldsymbol{v}_a^\top \boldsymbol{y})$ for some $a \in [nm]$. Generating a valid inequality accounts for finding a separating hyperplane that separates $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ from $\mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is a proper choice of breakpoints. For the rest of the paper, we choose the breakpoints $\boldsymbol{\beta}$ for a (2×2)-way disjunction in a specific manner, detailed in Construction 1.

Construction 1. *Consider a bounded convex relaxation $\tilde{\mathcal{K}}$ of $\mathcal{F}$. Let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \tilde{\mathcal{K}}$ be an optimal extreme point solution to the minimization problem over $\tilde{\mathcal{K}}$ such that $\boldsymbol{u}_a^\top \hat{\boldsymbol{W}}\boldsymbol{v}_a \neq (\boldsymbol{u}_a^\top \hat{\boldsymbol{x}})(\boldsymbol{v}_a^\top \hat{\boldsymbol{y}})$ for some $a \in [nm]$. We form a (2×2)-way disjunction of the form (7) on $\tilde{\mathcal{K}}$, based on $\boldsymbol{c}_a = (\boldsymbol{u}_a, \boldsymbol{v}_a)$ and $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$, using the following choice of the breakpoints, which are obtained by solving LPs:*

$$\begin{aligned}
&\beta_{1,1} = \min\{\tfrac{\boldsymbol{u}_a^\top \boldsymbol{x} + \boldsymbol{v}_a^\top \boldsymbol{y}}{2} \mid (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \tilde{\mathcal{K}}\}, \quad && \beta_{2,1} = \min\{\tfrac{\boldsymbol{u}_a^\top \boldsymbol{x} - \boldsymbol{v}_a^\top \boldsymbol{y}}{2} \mid (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \tilde{\mathcal{K}}\}, \\
&\beta_{1,2} = \tfrac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} + \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}, && \beta_{2,2} = \tfrac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} - \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}, \\
&\beta_{1,3} = \max\{\tfrac{\boldsymbol{u}_a^\top \boldsymbol{x} + \boldsymbol{v}_a^\top \boldsymbol{y}}{2} \mid (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \tilde{\mathcal{K}}\}, && \beta_{2,3} = \max\{\tfrac{\boldsymbol{u}_a^\top \boldsymbol{x} - \boldsymbol{v}_a^\top \boldsymbol{y}}{2} \mid (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \tilde{\mathcal{K}}\}.
\end{aligned}$$

Lemma 5 shows that the choice of breakpoints $\boldsymbol{\beta}$ based on Construction 1 leads to a single-vector disjunction that can be used to generate a valid disjunctive cut to cut off an extreme point $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \tilde{\mathcal{K}}$, where $\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a \neq (\boldsymbol{u}_a^\top \hat{\boldsymbol{x}})(\boldsymbol{v}_a^\top \hat{\boldsymbol{y}})$ for some $a \in [mn]$.

Lemma 5. *Consider a bounded convex relaxation $\tilde{\mathcal{K}}$ of $\mathcal{F}$. Let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \tilde{\mathcal{K}}$ be an extreme point solution such that $|\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \hat{\boldsymbol{x}})(\boldsymbol{v}_a^\top \hat{\boldsymbol{y}})| > \epsilon$ for some $\epsilon > 0$ and some $a \in [nm]$. Let $\vee_{k=1}^4 \tilde{S}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ be a (2×2)-way disjunction, where the breakpoints are chosen as in Construction 1 and using $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$. Then, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \notin \tilde{S}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ for all $k \in \{1,2,3,4\}$. Moreover, for all $k \in \{1,2,3,4\}$, one of the secant-induced inequalities in $\tilde{S}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$, defined in (8), is violated with an amount greater than ϵ by $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$.*

Proof. We first show that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \notin \tilde{S}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ for all $k \in \{1,2,3,4\}$, consequently, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \notin \vee_{k=1}^4 \tilde{S}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$. Suppose by contradiction that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \tilde{S}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ for some $\hat{k} \in \{1,2,3,4\}$. Without loss of generality, suppose that in $\tilde{S}_{\hat{k}}(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$, $\beta_{1,1}$ and $\beta_{1,3}$ denote the lower and upper bounds of $\frac{\boldsymbol{u}_a^\top \boldsymbol{x} + \boldsymbol{v}_a^\top \boldsymbol{y}}{2}$, respectively. Moreover, $\beta_{2,1}$ and $\beta_{2,3}$ denote the lower and upper bounds of $\frac{\boldsymbol{u}_a^\top \boldsymbol{x} - \boldsymbol{v}_a^\top \boldsymbol{y}}{2}$. Because $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \tilde{\mathcal{S}}_{\hat{k}}(\boldsymbol{c}, \tilde{\mathcal{K}}, \boldsymbol{\beta})$, we have

$$\begin{aligned} &\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a - \left(\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} + \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right)\left(\beta_{1,1} + \frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} + \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right) + \beta_{1,1}\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} + \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2} + \\ &\left(\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} - \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right)^2 = \boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a - \left(\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} + \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right)^2 + \left(\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} - \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right)^2 \leq 0. \end{aligned} \tag{14}$$

Note that if $\tilde{\mathcal{S}}_{\hat{k}}(\boldsymbol{c}, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ is such that $\beta_{1,2} \leq \frac{\boldsymbol{u}^\top \boldsymbol{x} + \boldsymbol{v}^\top \boldsymbol{y}}{2} \leq \beta_{1,3}$, we would still get a similar conclusion as in (14). So, the definition of $\tilde{\mathcal{S}}_{\hat{k}}(\boldsymbol{c}, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ as above is without loss of generality. With a similar argument, we conclude $-\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a + \left(\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} + \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right)^2 - \left(\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} - \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right)^2 \leq 0$. The above two inequalities imply that $\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a = (\boldsymbol{u}_a^\top \hat{\boldsymbol{x}})(\boldsymbol{v}_a^\top \hat{\boldsymbol{y}})$, yielding a contradiction.

Now, we show that for all $k \in \{1,2,3,4\}$, one of the secant inequalities in $\tilde{S}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ is violated by $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ and the amount of violation is greater than ϵ. First, suppose that $\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \hat{\boldsymbol{x}})(\boldsymbol{v}_a^\top \hat{\boldsymbol{y}}) > \epsilon$. Using the equality $(\boldsymbol{u}_a^\top \hat{\boldsymbol{x}})(\boldsymbol{v}_a^\top \hat{\boldsymbol{y}}) = \left(\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} + \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right)^2 - \left(\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} - \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right)^2$, we have $\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a - \left(\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} + \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right)^2 + \left(\frac{\boldsymbol{u}_a^\top \hat{\boldsymbol{x}} - \boldsymbol{v}_a^\top \hat{\boldsymbol{y}}}{2}\right)^2 > \epsilon$. The left-hand side of this inequality is the left-hand side of (14), implying that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ violates the first secant-induced inequality of $\tilde{S}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ for all $k \in \{1,2,3,4\}$, and the amount of violation is greater than ϵ.

Now, suppose that $-\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a + (\boldsymbol{u}_a^\top \hat{\boldsymbol{x}})(\boldsymbol{v}_a^\top \hat{\boldsymbol{y}}) > \epsilon$. Similarly, we conclude that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ violates the second secant-induced inequality of $\tilde{S}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ for all $k \in \{1,2,3,4\}$, and the amount of violation is greater than ϵ. ☐

3.3 Valid Disjunctive Cut and Projection Problem

So far, we have established our construction to choose the breakpoints $\boldsymbol{\beta}$. We now show that the choice of breakpoints $\boldsymbol{\beta}$ based on Construction 1 leads to a valid disjunctive cut. To obtain a valid cut for conv $(\mathcal{F})$ using a (2×2)-way disjunction, one may solve the corresponding CGLP, introduced

Algorithm 1 $\texttt{SepCuts}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}; \tilde{\mathcal{K}}, \boldsymbol{u}_a, \boldsymbol{v}_a)$

Input: $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$, $\tilde{\mathcal{K}}$, and $(\boldsymbol{u}_a, \boldsymbol{v}_a)$.

Output: $(\texttt{viol}, \boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{H}, \rho)$. If a valid inequality $\boldsymbol{\alpha}^\top \boldsymbol{x} + \boldsymbol{\theta}^\top \boldsymbol{y} + \boldsymbol{H} \bullet \boldsymbol{W} \geq \rho$ is found that is violated by $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$, then return viol=TRUE, $\boldsymbol{\alpha}$, $\boldsymbol{\theta}$, $\boldsymbol{H}$, and ρ. Otherwise, return viol=FALSE, $\boldsymbol{\alpha} = \mathbf{0}$, $\boldsymbol{\theta} = \mathbf{0}$, $\boldsymbol{H} = \mathbf{0}$, and $\rho = 0$.

Let $\boldsymbol{\alpha} = \mathbf{0}$, $\boldsymbol{\theta} = \mathbf{0}$, $\boldsymbol{H} = \mathbf{0}$, and $\rho = 0$.

Let $\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{v})$ and $\boldsymbol{\beta}$ is chosen as in Construction 1.

if $\tilde{\mathcal{S}}_k(\boldsymbol{c}, \tilde{\mathcal{K}}, \boldsymbol{\beta}) = \emptyset$ for all $k \in \{1, 2, 3, 4\}$ **then**

    viol $\leftarrow$ FALSE.

else

    Let $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{W}^*)$ be an optimal solution to $\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W}) \in \mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta})} \|(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) - (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})\|$.

    Let $\boldsymbol{\alpha}$, $\boldsymbol{\theta}$, and $\boldsymbol{H}$ be partial subgradients of $\|(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) - (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})\|$ at $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{W}^*)$ with respect to $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{W}$, respectively. Let $\rho = \boldsymbol{\alpha}^\top \boldsymbol{x}^* + \boldsymbol{\theta}^\top \boldsymbol{y}^* + \boldsymbol{H} \bullet \boldsymbol{W}^*$.

    viol $\leftarrow$ TRUE.

end if

in Lemma 4. As mentioned, this CGLP contains the outer approximation to the convex quadratic terms. Alternatively, one can solve a projection problem that minimizes the distance, measured by some norm, from $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ to a point in $\mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta})$.

Proposition 3. *Consider a bounded convex relaxation $\tilde{\mathcal{K}}$ of $\mathcal{F}$. Let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ be an optimal extreme point solution of $\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W}) \in \tilde{\mathcal{K}}} \; \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W}$. Suppose that $\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a \neq (\boldsymbol{u}_a^\top \hat{\boldsymbol{x}})(\boldsymbol{v}_a^\top \hat{\boldsymbol{y}})$ for some $a \in [nm]$. Moreover, let $\vee_{k=1}^4 \tilde{S}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$ be a (2×2)-way disjunction, where the breakpoints are chosen as in Construction 1 and using $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$. Furthermore, suppose that $\mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta})$, defined in (13), is nonempty. Then, the following projection problem*

$$\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W}) \in \mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta})} \|(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) - (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})\| \tag{15}$$

has a strictly positive and finite optimal value.

Proof. By Lemma 5, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \notin \tilde{\mathcal{S}}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$, $k \in \{1, 2, 3, 4\}$. Thus, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \notin \vee_{k=1}^4 \tilde{\mathcal{S}}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$. Because $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ is an extreme point of $\tilde{\mathcal{K}}$, it cannot be written as a convex combination of points in $\tilde{\mathcal{K}}$, including the points in $\vee_{k=1}^4 \tilde{\mathcal{S}}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$. Thus, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \notin \mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta})$. Because, $\mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta}) \neq \emptyset$, then, the optimal value to (15) is finite. Moreover, by Ruszczyński [26, Theorem 2.14], $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ can be strongly separated from $\mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta})$. By Rockafellar [24, Theorem 11.4], the strong separation holds if and only if the optimal value to (15) is strictly positive. □

An implication of Proposition 3 is that by choosing the breakpoints $\boldsymbol{\beta}$ according to Construction 1, one can separate $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ from $\mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta})$, and consequently, from $\operatorname{conv}(\mathcal{F})$. We summarize the cut generation procedure in Algorithm 1. In Section 4, we describe how these cutting planes can be utilized algorithmically to obtain an ϵ-optimal solution to (BLP) or to detect infeasibility in a finite number of iterations.

Remark 1. *The projection problem (15) is a minimum distance problem over set $\mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta})$. To* *obtain this set, one can first form a mixed-binary convex set whose projection onto the* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W})$*-space gives* $\vee_{k=1}^{4} \tilde{\mathcal{S}}_k(\boldsymbol{c}_a, \tilde{\mathcal{K}}, \boldsymbol{\beta})$*. Now, using a sequential convexification procedure, one can obtain the convex hull of this mixed-binary convex set (see Stubbs and Mehrotra [34, Proposition 1 and Theorem 1] for an illustration on general mixed-binary convex sets). By projecting the convex hull onto the* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W})$*, one can describe* $\mathcal{P}_a(\tilde{\mathcal{K}}, \boldsymbol{\beta})$*, given the fact that* $\mathrm{Proj}_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})}(\mathrm{conv}\,(\cdot)) = \mathrm{conv}\left(\mathrm{Proj}_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})}(\cdot)\right)$*. We skip the details for brevity and refer the readers to [34].*

4 A Finitely-Convergent Cutting Plane Algorithm

Motivated by Example 2 and numerical experiments in [9], in this section, a finitely-convergent cutting plane algorithm to obtain an ϵ-optimal solution to (BLP) or detect infeasibility is analyzed. The analyzed algorithm utilizes cuts obtained from the single-vector disjunction, described in Section 3.

A usual approach for such a cutting plane algorithm is to generate cuts by using one or more disjunctions obtained from *one* optimal solution to the current relaxation. For example, in the numerical experiments in [9], at most four cuts are generated, based on the largest four singular values of $\hat{\boldsymbol{W}} - \hat{\boldsymbol{x}}\hat{\boldsymbol{y}}^\top$, where $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ is an optimal solution to the current relaxation. In Section 2.2, we illustrated that such an approach is not sufficient to get arbitrarily close to the convex hull of solutions and obtain a globally optimal solution.

Unlike the usual cutting plane approaches that generate a valid inequality *only* at the current optimal solution, in this section, generating inequalities at *multiple* extreme point solutions of the current relaxation is analyzed algorithmically. These extreme points are generated by exploring near-optimal solutions to the current relaxation. The analyzed algorithm requires two input parameters $\gamma > 0$ and $\epsilon > 0$. The parameter γ determines the neighboring set of the current solution that the algorithm explores at each iteration. The parameter ϵ determines the optimality and feasibility tolerance.

In this section, we describe a modification to the cutting plane algorithm proposed in [9] to generate cuts at multiple vertices of the current relaxation. In Section 4.2, a finitely-convergent algorithm is analyzed when two sets of bases for $\mathbb{R}^n$ and $\mathbb{R}^m$ are available a priori. In Section 4.3, we suppose such bases are unavailable a priori and obtained through SVD.

4.1 Definitions and Technical Results

In this section, we give definitions of an ϵ-feasible solution and ϵ-optimal solution to (BLP), and present some technical results.

Definition 1. *(**ϵ-Feasible Solution**) For the optimization problem* (BLP) *with* $z^* = \min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\mathcal{F}} \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W}$*, we say that a point* $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \mathcal{K}$ *is ϵ-feasible if* $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \mathcal{F}^\epsilon$*, as defined in* (12)*. In other words,* $|\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \hat{\boldsymbol{x}})(\boldsymbol{v}_a^\top \hat{\boldsymbol{y}})| \le \epsilon$ *for all* $a \in [mn]$*.*

Note that by Proposition 2, an ϵ-feasible solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ also belongs to $\bar{\mathcal{F}}^\epsilon$, as defined in (11). Consequently, $|\boldsymbol{u}_a^\top \hat{\boldsymbol{W}} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \hat{\boldsymbol{x}})(\boldsymbol{v}_a^\top \hat{\boldsymbol{y}})| \le mn\epsilon$.

Definition 2. *(ϵ-**Optimal Solution**) For the optimization problem (BLP) with $z^* = \min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\mathcal{F}}$ $\boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W}$, we say that a point $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \mathcal{K}$ is an ϵ-optimal solution if $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ is ϵ-feasible and $\boldsymbol{f}_0^\top \hat{\boldsymbol{x}} + \boldsymbol{g}_0^\top \hat{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \hat{\boldsymbol{W}} \le z^* + \epsilon$.*

We now state some technical results about a convergent sequence of sets that we will use in the sequel. Lemma 6 states that the order of convex hull and limit operators can be exchanged. Lemma 7 states every extreme point of the limiting set is an accumulation point of a sequence of extreme points.

Lemma 6. *Let $\{\mathcal{B}_1^t\}, \{\mathcal{B}_2^t\}, \ldots, \{\mathcal{B}_\kappa^t\}$ be convergent sequences of nonempty compact connected sets of a finite-dimensional Euclidean space. If $\text{h-lim}_{t\to\infty} \mathcal{B}_\iota^t = \bar{\mathcal{B}}_\iota$, where $\bar{\mathcal{B}}_\iota$ is nonempty, for $\iota \in [\kappa]$, then,*

$$\underset{t\to\infty}{\text{h-lim}}\, \text{conv}\left(\cup_{\iota=1}^\kappa \mathcal{B}_\iota^t\right) = \text{conv}\left(\cup_{\iota=1}^\kappa \underset{t\to\infty}{\text{h-lim}}\, \mathcal{B}_\iota^t\right) = \text{conv}\left(\cup_{\iota=1}^\kappa \bar{\mathcal{B}}_\iota\right).$$

Proof. We show that for any $\delta > 0$, there exists $\hat{t} > 0$ such that for all $t \ge \hat{t}$, we have $d_{\text{H}}\Big(\text{conv}\left(\cup_{\iota=1}^\kappa \mathcal{B}_\iota^t\right), \text{conv}\left(\cup_{\iota=1}^\kappa \bar{\mathcal{B}}_\iota\right)\Big) \le \delta$. In other words, for all $t \ge \hat{t}$, we have $\min_{\boldsymbol{b}\in\text{conv}\left(\cup_{\iota=1}^\kappa \bar{\mathcal{B}}_\iota\right)} \|\boldsymbol{b} - \boldsymbol{b}^t\| \le \delta$ for all $\boldsymbol{b}^t \in \text{conv}\left(\cup_{\iota=1}^\kappa \mathcal{B}_\iota^t\right)$ and $\min_{\boldsymbol{b}^t\in\text{conv}\left(\cup_{\iota=1}^\kappa \mathcal{B}_\iota^t\right)} \|\boldsymbol{b} - \boldsymbol{b}^t\| \le \delta$ for all $\boldsymbol{b} \in \text{conv}\left(\cup_{\iota=1}^\kappa \bar{\mathcal{B}}_\iota\right)$.

First, note that because $\text{h-lim}_{t\to\infty} \mathcal{B}_\iota^t = \bar{\mathcal{B}}_\iota$, then, for any $\delta > 0$, there exists $\hat{t}_\iota > 0$ such that for all $t \ge \hat{t}_\iota$, we have $\min_{\boldsymbol{b}_\iota\in\bar{\mathcal{B}}_\iota} \|\boldsymbol{b}_\iota - \boldsymbol{b}_\iota^t\| \le \delta$ for all $\boldsymbol{b}_\iota^t \in \mathcal{B}_\iota^t$ and $\min_{\boldsymbol{b}_\iota^t\in\mathcal{B}_\iota^t} \|\boldsymbol{b}_\iota - \boldsymbol{b}_\iota^t\| \le \delta$ for all $\boldsymbol{b}_\iota \in \bar{\mathcal{B}}_\iota$. Moreover, for any $t > 0$, $\boldsymbol{b}^t \in \text{conv}\left(\cup_{\iota=1}^\kappa \mathcal{B}_\iota^t\right)$ can be written as $\boldsymbol{b}^t = \sum_{\iota=1}^\kappa \lambda_\iota^t \boldsymbol{b}_\iota^t$ for some $\boldsymbol{b}_\iota^t \in \mathcal{B}_\iota^t$ and $\lambda_\iota^t \in [0,1]$, $\iota \in [\kappa]$, such that $\sum_{\iota=1}^\kappa \lambda_\iota^t = 1$. Therefore,

$$\begin{aligned}
&\min_{\boldsymbol{b}\in\text{conv}\left(\cup_{\iota=1}^\kappa \bar{\mathcal{B}}_\iota\right)} \|\boldsymbol{b} - \boldsymbol{b}^t\| = \min_{\boldsymbol{b}\in\text{conv}\left(\cup_{\iota=1}^\kappa \bar{\mathcal{B}}_\iota\right)} \Big\| \sum_{\iota=1}^\kappa \lambda_\iota^t (\boldsymbol{b} - \boldsymbol{b}_\iota^t) \Big\| \\
&\le \min_{\boldsymbol{b}_\iota\in\bar{\mathcal{B}}_\iota,\, \iota\in[\kappa]} \Big\| \sum_{\iota=1}^\kappa \lambda_\iota^t (\boldsymbol{b}_\iota - \boldsymbol{b}_\iota^t) \Big\| \le \min_{\boldsymbol{b}_\iota\in\bar{\mathcal{B}}_\iota,\, \iota\in[\kappa]} \sum_{\iota=1}^\kappa \lambda_\iota^t \|\boldsymbol{b}_\iota - \boldsymbol{b}_\iota^t\| = \sum_{\iota=1}^\kappa \lambda_\iota^t \min_{\boldsymbol{b}_\iota\in\bar{\mathcal{B}}_\iota} \|\boldsymbol{b}_\iota - \boldsymbol{b}_\iota^t\|,
\end{aligned}$$

where the first inequality follows because $\{\boldsymbol{b} = \sum_{\iota=1}^\kappa \lambda_\iota^t \boldsymbol{b}_\iota \mid \boldsymbol{b}_\iota \in \bar{\mathcal{B}}_\iota,\ \iota \in [\kappa]\} \subseteq \text{conv}\left(\cup_{\iota=1}^\kappa \bar{\mathcal{B}}_\iota\right)$. By choosing $\hat{t} := \max_{\iota=1}^\kappa \hat{t}_\iota$, it follows that $\min_{\boldsymbol{b}\in\text{conv}\left(\cup_{\iota=1}^\kappa \bar{\mathcal{B}}_\iota\right)} \|\boldsymbol{b} - \boldsymbol{b}^t\| \le \delta$ for all $\boldsymbol{b}^t \in \text{conv}\left(\cup_{\iota=1}^\kappa \mathcal{B}_\iota^t\right)$.

Similarly, any $\boldsymbol{b} \in \text{conv}\left(\cup_{\iota=1}^\kappa \bar{\mathcal{B}}_\iota\right)$ can be written as $\boldsymbol{b} = \sum_{\iota=1}^\kappa \lambda_\iota \boldsymbol{b}_\iota$ for some $\boldsymbol{b}_\iota \in \bar{\mathcal{B}}_\iota$ and $\lambda_\iota \in [0,1]$, $\iota = 1, \ldots, \kappa$, such that $\sum_{\iota=1}^\kappa \lambda_\iota = 1$. Therefore,

$$\begin{aligned}
&\min_{\boldsymbol{b}^t\in\text{conv}\left(\cup_{\iota=1}^\kappa \mathcal{B}_\iota^t\right)} \|\boldsymbol{b} - \boldsymbol{b}^t\| = \min_{\boldsymbol{b}^t\in\text{conv}\left(\cup_{\iota=1}^\kappa \mathcal{B}_\iota^t\right)} \Big\| \sum_{\iota=1}^\kappa \lambda_\iota (\boldsymbol{b}_\iota - \boldsymbol{b}^t) \Big\| \\
&\le \min_{\boldsymbol{b}_\iota^t\in\mathcal{B}_\iota^t,\, \iota\in[\kappa]} \Big\| \sum_{\iota=1}^\kappa \lambda_\iota (\boldsymbol{b}_\iota - \boldsymbol{b}_\iota^t) \Big\| \le \min_{\boldsymbol{b}_\iota^t\in\mathcal{B}_\iota^t,\, \iota\in[\kappa]} \sum_{\iota=1}^\kappa \lambda_\iota \|\boldsymbol{b}_\iota - \boldsymbol{b}_\iota^t\| = \sum_{\iota=1}^\kappa \lambda_\iota^t \min_{\boldsymbol{b}_\iota^t\in\mathcal{B}_\iota^t} \|\boldsymbol{b}_\iota - \boldsymbol{b}_\iota^t\|,
\end{aligned}$$

where the first inequality follows due $\{\boldsymbol{b}^t = \sum_{\iota=1}^\kappa \lambda_\iota \boldsymbol{b}_\iota^t \mid \boldsymbol{b}_\iota^t \in \mathcal{B}_\iota^t,\ \iota = 1, \ldots, \kappa\} \subseteq \text{conv}\left(\cup_{\iota=1}^\kappa \mathcal{B}_\iota^t\right)$. By choosing $\hat{t} := \max_{\iota=1}^\kappa \hat{t}_\iota$, it follows that $\min_{\boldsymbol{b}^t\in\text{conv}\left(\cup_{\iota=1}^\kappa \mathcal{B}_\iota^t\right)} \|\boldsymbol{b} - \boldsymbol{b}^t\| \le \delta$ for all $\boldsymbol{b} \in \text{conv}\left(\cup_{\iota=1}^\kappa \bar{\mathcal{B}}_\iota\right)$.

This completes the proof. $\square$

Lemma 7. *Owen and Mehrotra [21, Lemma 2] Let $\{\mathcal{B}^t\}$ be a convergent sequence of bounded convex sets such that $\mathcal{B}^{t+1} \subseteq \mathcal{B}^t$ for all $t \geq 0$ and $\text{h-lim}_{t\to\infty} \mathcal{B}^t = \tilde{\mathcal{B}}$. For each $\tilde{b} \in \text{ext}(\tilde{\mathcal{B}})$, there exists some sequence $\{b^t\}$ of points in $\text{ext}(\mathcal{B}^t)$ with a subsequence converging to $\tilde{b}$.*

4.2 General Basis

In Algorithm 2, a finitely-convergent algorithm is analyzed when two sets of bases for $\mathbb{R}^n$ and $\mathbb{R}^m$ are available. By relying on the cut-generation procedure, described in Algorithm 1, this algorithm either generates an ϵ-optimal solution to (BLP) or detects its infeasibility (i.e., $\mathcal{F} = \emptyset$) in a finite number of iterations (see Theorem 1).

Before we proceed, let us introduce the notation we use in Algorithm 2. At each iteration t, we denote the current relaxation by S^t. Let $\underline{z}^t = \min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in S^t} \ \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W}$. Recall that the input parameter $\gamma > 0$ controls the vertex exploration in the neighborhood of the current solution. We impose the vertex exploration by defining the set of extreme point solutions whose objective values are γ-away from $\underline{z}^t$ as

$$\Omega^t := \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \text{ext}(S^t) \big| \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W} - \underline{z}^t \leq \gamma\}. \tag{16}$$

Given $\epsilon > 0$, we define a subset of ϵ-feasible solutions in Ω^t as

$$\Omega_\epsilon := \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \Omega^t \big| |\boldsymbol{u}_a^\top \boldsymbol{W} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \boldsymbol{x})(\boldsymbol{v}_a^\top \boldsymbol{y})| \leq \epsilon, \ a \in [nm]\}. \tag{17}$$

As it can be seen from Algorithm 2, the algorithm proposed in [9] is modified to generate disjunctive cuts at all points $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{W}}) \in \Omega^t$ and $a \in [nm]$ such that $|\boldsymbol{u}_a^\top \bar{\boldsymbol{W}} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \bar{\boldsymbol{x}})(\boldsymbol{v}_a^\top \bar{\boldsymbol{y}})| > \frac{\epsilon}{4}$. We show that the analyzed vertex exploration, imposed by $\gamma > 0$, enables the algorithm to generate cuts that are deep enough to cut off points that are not in $\text{conv}(\mathcal{F})$, and eventually lead to an ϵ-optimal solution to (BLP) or detects infeasibility. Before we state and prove the main result of this section, we address the building blocks of Algorithm 2 and state some intermediate results. In particular, Lemma 8 shows that Algorithm 2 explores extreme point solutions in Ω^t. Lemma 10 shows that cuts generated at these extreme point solutions have a sufficiently large depth to cut off points that violate constraints in $\mathcal{F}$.

Lemma 8. *Let $\{S^t\}$ be a sequence of nonempty sets generated by Algorithm 2. Suppose that $\{S^t\}$ converges to a nonempty set $\tilde{S}$, and let $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$ be an optimal extreme point solution of $\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in \tilde{S}} \ \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W}$. Let $\{(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t)\}_{t\in\mathcal{T}}$ be a convergent subsequence of points in $\text{ext}(S^t)$ such that $\{(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t)\}_{t\in\mathcal{T}} \to (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$. Then, for any $\gamma > 0$, there exists a sufficiently large $t \in \mathcal{T}$ such that $(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) \in \Omega^t$, where Ω^t is defined in (16).*

Before we prove Lemma 8, several remarks are in order. The extreme point solutions $\{(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t)\}_{t\in\mathcal{T}} \in \text{ext}(S^t)$ are not necessarily an optimal solution to $\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in S^t} \ \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} +$

$\boldsymbol{A}_0 \bullet \boldsymbol{W}$. Hence, one should not expect the lemma statement to hold necessarily when $\gamma = 0$. The key piece in Lemma 8 is choosing γ to be strictly positive (see the subsequent limiting arguments in the proof of Theorem 1). In fact, when $\gamma = 0$, there would be no neighboring set, and hence, no vertex exploration takes place. On the other hand, any $\gamma > 0$ implies the vertex exploration. Nevertheless, this does not imply that Ω^t is nonsingleton for any t. Indeed, it is theoretically possible to have $|\Omega^t| = 1$ (with the optimal solution at the t-th iteration to be the only element in this set), but this can only happen in a finite number of iterations. As stated in Lemma 8, for any $\gamma > 0$, there exists a sufficiently large t such that Ω^t is eventually nonsingleton. The choice of $\gamma > 0$, and hence the vertex exploration, is a conceptual modification to the algorithm investigated in [9]. We shall shortly see in the proof of Theorem 1 that the vertex exploration plays a role in the finite-convergence analysis of Algorithm 2.

Proof of Lemma 8. First, because $S^0 (= \mathcal{K})$ is bounded and $S^{t+1} \subseteq S^t$ for all $t > 0$, then the sequence of nonempty compact sets $\{S^t\}$ converges to a set $\tilde{S}$ by Lemma 2. Furthermore, by Lemma 7, there exists a convergent subsequence $\{(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t)\}_{t \in \mathcal{T}}$ of points in $\text{ext}(S^t)$ such that $\{(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t)\}_{t \in \mathcal{T}} \to (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$. Now, note that $\boldsymbol{f}_0^\top \boldsymbol{x}^t + \boldsymbol{g}_0^\top \boldsymbol{y}^t + \boldsymbol{A}_0 \bullet \boldsymbol{W}^t - \underline{z}^t = \boldsymbol{f}_0^\top \boldsymbol{x}^t + \boldsymbol{g}_0^\top \boldsymbol{y}^t + \boldsymbol{A}_0 \bullet \boldsymbol{W}^t - (\boldsymbol{f}_0^\top \tilde{\boldsymbol{x}} + \boldsymbol{g}_0^\top \tilde{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \tilde{\boldsymbol{W}}) + (\boldsymbol{f}_0^\top \tilde{\boldsymbol{x}} + \boldsymbol{g}_0^\top \tilde{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \tilde{\boldsymbol{W}}) - \underline{z}^t$. By the fact that $\{(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t)\}_{t \in \mathcal{T}} \to (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$ and the continuity of $\boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W}$, there exists t_1 such that for $t \geq t_1$, $t \in \mathcal{T}$, we have $\boldsymbol{f}_0^\top \boldsymbol{x}^t + \boldsymbol{g}_0^\top \boldsymbol{y}^t + \boldsymbol{A}_0 \bullet \boldsymbol{W}^t - (\boldsymbol{f}_0^\top \tilde{\boldsymbol{x}} + \boldsymbol{g}_0^\top \tilde{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \tilde{\boldsymbol{W}}) \leq \frac{\gamma}{2}$. Moreover, because $\text{h-lim}_{t \to \infty} S^t = \tilde{S}$, we have $\{\underline{z}^t\} \to \boldsymbol{f}_0^\top \tilde{\boldsymbol{x}} + \boldsymbol{g}_0^\top \tilde{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \tilde{\boldsymbol{W}}$. This implies that there exists t_2 such that for $t \geq t_2$, $t \in \mathcal{T}$, we have $\boldsymbol{f}_0^\top \tilde{\boldsymbol{x}} + \boldsymbol{g}_0^\top \tilde{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \tilde{\boldsymbol{W}} - \underline{z}^t \leq \frac{\gamma}{2}$. Consequently, for $t \geq \max\{t_1, t_2\}$, $t \in \mathcal{T}$, we have $\boldsymbol{f}_0^\top \boldsymbol{x}^t + \boldsymbol{g}_0^\top \boldsymbol{y}^t + \boldsymbol{A}_0 \bullet \boldsymbol{W}^t - \underline{z}^t \leq \gamma$, which implies that $(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) \in \Omega^t$. □

Lemma 9. *Consider the assumptions and notation in Lemma 8.*

(i) If $|\boldsymbol{u}_a^\top \tilde{\boldsymbol{W}} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \tilde{\boldsymbol{x}})(\boldsymbol{v}_a^\top \tilde{\boldsymbol{y}})| > \epsilon$ for some $a \in [nm]$, then, there exists a sufficiently large $t \in \mathcal{T}$ such that $|\boldsymbol{u}_a^\top \boldsymbol{W}^t \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \boldsymbol{x}^t)(\boldsymbol{v}_a^\top \boldsymbol{y}^t)| > \frac{\epsilon}{2}$.

(ii) If $|\boldsymbol{u}_a^\top \tilde{\boldsymbol{W}} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \tilde{\boldsymbol{x}})(\boldsymbol{v}_a^\top \tilde{\boldsymbol{y}})| \leq \epsilon$ for all $a \in [nm]$, then, there exists a sufficiently large $t \in \mathcal{T}$ such that $|\boldsymbol{u}_a^\top \boldsymbol{W}^t \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \boldsymbol{x}^t)(\boldsymbol{v}_a^\top \boldsymbol{y}^t)| \leq 2\epsilon$ for all $a \in [nm]$.

Proof. The proof is immediate from the continuity of $\boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W}$, and we skip it for brevity. □

Lemma 10 shows that there exists a sufficiently large t such that the valid inequality generated at $(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) \in \Omega^t$ have a sufficiently large depth and subsequently, can cut off point $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$, with $|\boldsymbol{u}_a^\top \tilde{\boldsymbol{W}} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \tilde{\boldsymbol{x}})(\boldsymbol{v}_a^\top \tilde{\boldsymbol{y}})| > \epsilon$ for some $a \in [nm]$, from S^t.

Lemma 10. *Consider the assumptions in Lemma 9(i). Let the breakpoints $\tilde{\boldsymbol{\beta}}$ be chosen as in Construction 1, for $\tilde{S}$ and point $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$. Then, we have*

$$\delta := \min_{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \in \mathcal{P}_a(\tilde{S}, \tilde{\boldsymbol{\beta}})} \|(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) - (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W})\| > 0.$$

Furthermore, there exists a sufficiently large $t \in \mathcal{T}$ such that

$$\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\mathcal{P}_a(S^t,\boldsymbol{\beta}^t)} \|(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{W}^t)-(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\| > \frac{\delta}{2},$$

where the breakpoints $\boldsymbol{\beta}^t$ are chosen as in Construction 1, for S^t and $(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{W}^t)$.

To prove Lemma 10, we first show in Lemma 11 that there exists a sufficiently large t such that point $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{\boldsymbol{W}})$, introduced in Lemma 8, with $|\boldsymbol{u}_a^\top\tilde{\boldsymbol{W}}\boldsymbol{v}_a-(\boldsymbol{u}_a^\top\tilde{\boldsymbol{x}})(\boldsymbol{v}_a^\top\tilde{\boldsymbol{y}})|>\epsilon$ for some $a\in[nm]$, violates the disjunction formed over S^t and using the breakpoints $\boldsymbol{\beta}^t$, chosen as in Construction 1 for point $(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{W}^t)$.

Lemma 11. *Consider the assumptions in Lemma 9(i). Let the breakpoints $\boldsymbol{\beta}^t$ be chosen as in Construction 1, for S^t and point $(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{W}^t)$. Then, there exists a sufficiently large $t\in\mathcal{T}$ such that for all $k\in\{1,2,3,4\}$, one of the secant-induced inequalities in $\tilde{S}_k(\boldsymbol{c}_a,S^t,\boldsymbol{\beta}^t)$, defined in (8), is violated by $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{\boldsymbol{W}})$, and the amount of violation is greater than $\frac{\epsilon}{2}$.*

Proof. Suppose that $\boldsymbol{u}_a^\top\tilde{\boldsymbol{W}}\boldsymbol{v}_a-(\boldsymbol{u}_a^\top\tilde{\boldsymbol{x}})(\boldsymbol{v}_a^\top\tilde{\boldsymbol{y}})>\epsilon$. We first analyze the violation of the first secant-induced inequality in $\tilde{S}_k(\boldsymbol{c}_a,S^t,\boldsymbol{\beta}^t)$ by $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{\boldsymbol{W}})$ for all k, $k\in\{1,2,3,4\}$. Let us begin with those $\tilde{S}_k(\boldsymbol{c}_a,S^t,\boldsymbol{\beta}^t)$ for which $\beta_{1,1}^t$ is the lower bound on $\frac{\boldsymbol{u}_a^\top\boldsymbol{x}+\boldsymbol{v}_a^\top\boldsymbol{y}}{2}$. With some algebra, we have

$$\begin{aligned}
&\boldsymbol{u}_a^\top\tilde{\boldsymbol{W}}\boldsymbol{v}_a-\left(\frac{\boldsymbol{u}_a^\top\tilde{\boldsymbol{x}}+\boldsymbol{v}_a^\top\tilde{\boldsymbol{y}}}{2}\right)\left(\beta_{1,1}^t+\frac{\boldsymbol{u}_a^\top\boldsymbol{x}^t+\boldsymbol{v}_a^\top\boldsymbol{y}^t}{2}\right)+\beta_{1,1}^t\frac{\boldsymbol{u}_a^\top\boldsymbol{x}^t+\boldsymbol{v}_a^\top\boldsymbol{y}^t}{2}\\
&+\left(\frac{\boldsymbol{u}_a^\top\tilde{\boldsymbol{x}}-\boldsymbol{v}_a^\top\tilde{\boldsymbol{y}}}{2}\right)^2=\boldsymbol{u}_a^\top\tilde{\boldsymbol{W}}\boldsymbol{v}_a-(\boldsymbol{u}_a^\top\tilde{\boldsymbol{x}})(\boldsymbol{v}_a^\top\tilde{\boldsymbol{y}})\\
&+\left(\beta_{1,1}^t-\frac{\boldsymbol{u}_a^\top\tilde{\boldsymbol{x}}+\boldsymbol{v}_a^\top\tilde{\boldsymbol{y}}}{2}\right)\left(\frac{\boldsymbol{u}_a^\top(\boldsymbol{x}^t-\tilde{\boldsymbol{x}})+\boldsymbol{v}_a^\top(\boldsymbol{y}^t-\tilde{\boldsymbol{y}})}{2}\right)\\
>&\epsilon+\left(\beta_{1,1}^t-\frac{\boldsymbol{u}_a^\top\tilde{\boldsymbol{x}}+\boldsymbol{v}_a^\top\tilde{\boldsymbol{y}}}{2}\right)\left(\frac{\boldsymbol{u}_a^\top(\boldsymbol{x}^t-\tilde{\boldsymbol{x}})+\boldsymbol{v}_a^\top(\boldsymbol{y}^t-\tilde{\boldsymbol{y}})}{2}\right).
\end{aligned}\tag{18}$$

Observe that for $t>0$, we have $\beta_{1,1}^t-\frac{\boldsymbol{u}_a^\top\tilde{\boldsymbol{x}}+\boldsymbol{v}_a^\top\tilde{\boldsymbol{y}}}{2}\le 0$ because $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{\boldsymbol{W}})$ is suboptimal to $\beta_{1,1}^t=\min\{\frac{\boldsymbol{u}_a^\top\boldsymbol{x}+\boldsymbol{v}_a^\top\boldsymbol{y}}{2}\mid(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in S^t\}$. On the other hand, by the equivalence of the norms on a finite-dimensional vector space, we have $\frac{\boldsymbol{u}_a^\top(\boldsymbol{x}^t-\tilde{\boldsymbol{x}})+\boldsymbol{v}_a^\top(\boldsymbol{y}^t-\tilde{\boldsymbol{y}})}{2}<C\|(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{W}^t)-(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{\boldsymbol{W}})\|$ for some positive constant C. Hence, there exists t_1 such that for $t\ge t_1$, $t\in\mathcal{T}$, we have $C\left(\beta_{1,1}^t-\frac{\boldsymbol{u}_a^\top\tilde{\boldsymbol{x}}+\boldsymbol{v}_a^\top\tilde{\boldsymbol{y}}}{2}\right)\|(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{W}^t)-(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\tilde{\boldsymbol{W}})\|>-\frac{\epsilon}{2}$. Thus, the violation of the inequality (18) is greater than $\frac{\epsilon}{2}$. With a similar argument, we can conclude that for those $\tilde{S}_k(\boldsymbol{c}_a,S^t,\boldsymbol{\beta}^t)$ for which $\beta_{1,3}^t$ is the upper bound on $\frac{\boldsymbol{u}_a^\top\boldsymbol{x}+\boldsymbol{v}_a^\top\boldsymbol{y}}{2}$, the violation of the first secant-induced inequality is greater than $\frac{\epsilon}{2}$.

For the case that $-\boldsymbol{u}_a^\top\tilde{\boldsymbol{W}}\boldsymbol{v}_a+(\boldsymbol{u}_a^\top\tilde{\boldsymbol{x}})(\boldsymbol{v}_a^\top\tilde{\boldsymbol{y}})>\epsilon$, we can similarly conclude that the violation of the second secant-induced inequality is greater than $\frac{\epsilon}{2}$. Consequently, the result follows. ☐

Proof of Lemma 10. The first part follows from a direct application of Proposition 3. To prove the second part, let $\hat{t}_1$ be the sufficiently large t for which the result in Lemma 11 holds. Note that with a similar argument as in the proof of Lemma 11, we can find a sequence of sufficiently large t, $t\in\mathcal{T}$,

Algorithm 2 Disjunctive cutting plane for (BLP) using general bases

Input: $\mathcal{K}$, $\{\boldsymbol{u}_i\}_{i=1}^n$, $\{\boldsymbol{v}_j\}_{j=1}^m$, γ, and ϵ.
Output: An ϵ-optimal solution.
Set $t \leftarrow 0$ and $S^0 = \mathcal{K}$.
while $S^t \neq \emptyset$ **do**
    Let $\underline{z}^t$ be the optimal value of $\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in S^t} \ \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W}$.
    Let $\Omega^t := \{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W}) \in \operatorname{ext}(S^t) \,|\, \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W} - \underline{z}^t \leq \gamma\}$.
    Let $\Omega_\epsilon := \{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W}) \in \Omega^t \,|\, |\boldsymbol{u}_a^\top \boldsymbol{W} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \boldsymbol{x})(\boldsymbol{v}_a^\top \boldsymbol{y})| \leq \epsilon,\ a \in [nm]\}$.
    if $\Omega_\epsilon \neq \emptyset$ **then**
        for each $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \Omega_\epsilon$ **do**
            if $\boldsymbol{f}_0^\top \hat{\boldsymbol{x}} + \boldsymbol{g}_0^\top \hat{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \hat{\boldsymbol{W}} - \underline{z}^t \leq \epsilon$ **then**
                STOP and output $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ as an ϵ-optimal solution.
            end if
        end for
    end if
    $S^{t+1} = S^t$.
    for each $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{W}}) \in \Omega^t$ and $a \in [nm]$ such that $|\boldsymbol{u}_a^\top \bar{\boldsymbol{W}} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \bar{\boldsymbol{x}})(\boldsymbol{v}_a^\top \bar{\boldsymbol{y}})| > \frac{\epsilon}{4}$ **do**
        Call the procedure $\texttt{SepCuts}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{W}}; S^t, \boldsymbol{u}_a, \boldsymbol{v}_a)$ to obtain $(\texttt{viol}, \boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{H}, \rho)$.
        if $\texttt{viol}$=FALSE **then**
            STOP.
        else
            Let $S^{t+1} := \{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W}) \in S^{t+1} \,|\, \boldsymbol{\alpha}^\top \boldsymbol{x} + \boldsymbol{\theta}^\top \boldsymbol{y} + \boldsymbol{H} \bullet \boldsymbol{W} \geq \rho\}$.
        end if
    end for
    Set $t \leftarrow t+1$.
end while
STOP.

namely $\mathcal{T}' = \{\hat{t}_1, \hat{t}_2, \hat{t}_3, \ldots\}$, where for all $k \in \{1,2,3,4\}$, one of the secant-induced inequalities in $\tilde{S}_k(\boldsymbol{c}_a, S^t, \boldsymbol{\beta}^t)$ is violated by $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$ and the amount of violation is greater than $\{\epsilon - \frac{\epsilon}{2}, \epsilon - \frac{\epsilon}{4}, \epsilon - \frac{\epsilon}{8}, \ldots\}$. Hence, this subsequence $\mathcal{T}'$ yields a subsequence $\{S^t\}_{t\in\mathcal{T}'}$, for which $\vee_{k=1}^4 \tilde{S}_k(\boldsymbol{c}_a, S^t, \boldsymbol{\beta}^t)$ is a decreasing sequence. This is because $\{S^t\}$ is a decreasing sequence of nested sets and the amount of violation of one of the secant-induced inequalities in $\tilde{S}_k(\boldsymbol{c}_a, S^t, \boldsymbol{\beta}^t)$, $t \in \mathcal{T}'$, increases for all $k \in \{1,2,3,4\}$. Consequently, for the subsequence $\mathcal{T}'$, $\mathcal{P}_a(S^t, \boldsymbol{\beta}^t)$ is a decreasing sequence. On the other hand, $\{(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t)\}_{t\in\mathcal{T}} \to (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$ and $\text{h-lim}_{t\to\infty} S^t = \tilde{S}$. These imply that $\{\boldsymbol{\beta}^t\}_{t\in\mathcal{T}} \to \tilde{\boldsymbol{\beta}}$. Putting these all together, we conclude that $\{\mathcal{P}_a(S^t, \boldsymbol{\beta}^t)\}_{t\in\mathcal{T}'} \to \mathcal{P}_a(\tilde{S}, \tilde{\boldsymbol{\beta}})$ by Lemma 6. Thus, we have $\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\mathcal{P}_a(S^t,\boldsymbol{\beta}^t)} \|(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) - (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W})\| > -\frac{\delta}{2} + \min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\mathcal{P}_a(\tilde{S},\tilde{\boldsymbol{\beta}})} \|(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) - (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W})\| = \frac{\delta}{2}$ for a sufficiently large $t \in \mathcal{T}'$. This completes the proof. □

We are now ready to state the main result of this section for any general bases for $\mathbb{R}^n$ and $\mathbb{R}^m$, including the standard bases $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ and $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m\}$.

Theorem 1. *Consider two parameters $\gamma > 0$ and $\epsilon > 0$. Let $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$ denote a set of mutually orthonormal vectors in $\mathbb{R}^n$, and $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ denote a set of mutually orthonormal vectors in $\mathbb{R}^m$. Then, Algorithm 2 either generates an ϵ-optimal solution to (BLP) or detects infeasibility in a finite number of iterations.*

Proof. Consider the notation defined in the description of Algorithm 2. Note that the algorithm generates cutting planes for conv $(\mathcal{F})$. Suppose by contradiction that the algorithm does not converge in a finite number of iterations. Because $S^0(=\mathcal{K})$ is bounded and $S^{t+1} \subseteq S^t$ for all $t > 0$, then the sequence of closed sets $\{S^t\}$ converges to a set $\tilde{S}$ by Lemma 2. We examine the cases that $\tilde{S} = \emptyset$ and $\tilde{S} \neq \emptyset$.

Case 1. $\tilde{S} = \emptyset$. We conclude that $\mathcal{F}$ is empty because $\mathcal{F} \subseteq \text{conv}(\mathcal{F}) \subseteq \tilde{S}$. On the other hand, because $\{S^t\}$ converges to the empty set $\tilde{S}$, then, there exists a finite $t > 0$ such that $S^t = \emptyset$. Otherwise, $\tilde{S} = \cap_{t=1}^{\infty} S^t \neq \emptyset$ by the Cantor's intersection theorem, stated in Lemma 1. Hence, the algorithm terminates at line 26 of Algorithm 2 after detecting the infeasibility of (BLP).

Case 2. $\tilde{S} \neq \emptyset$. Let $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$ be an optimal extreme point solution of $\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\tilde{S}} \ \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W}$. By Lemma 7, there exists a convergent subsequence $\{(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t)\}_{t\in\mathcal{T}}$ of points in $\text{ext}(S^t)$ such that $\{(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t)\}_{t\in\mathcal{T}} \to (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$. We examine the cases that $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) \notin \mathcal{F}^{\frac{\epsilon}{2}}$ and $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) \in \mathcal{F}^{\frac{\epsilon}{2}}$ separately, where $\mathcal{F}^{\frac{\epsilon}{2}}$ is defined in (12).

Case 2.1. $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) \notin \mathcal{F}^{\frac{\epsilon}{2}}$. In this case, there exists some a, $a \in [nm]$, such that $|\boldsymbol{u}_a^\top \tilde{\boldsymbol{W}} \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \tilde{\boldsymbol{x}})(\boldsymbol{v}_a^\top \tilde{\boldsymbol{y}})| > \frac{\epsilon}{2}$. Let us choose $\tilde{\boldsymbol{\beta}}$ as in Construction 1 for $\tilde{S}$ and $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$. Let us similarly define $\boldsymbol{\beta}^t$ for S^t and $(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t)$. By the first part of Lemma 10, there exists $\delta > 0$, where δ is defined as follows:

$$\delta = \min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\mathcal{P}_a(\tilde{S},\tilde{\boldsymbol{\beta}})} \|(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) - (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W})\|.$$

Claim 1. *There exists a finite $t \in \mathcal{T}$ such that*

1. $\|(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) - (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})\| < \frac{\delta}{2}$ *(Lemma 7),*
2. $(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) \in \Omega^t$ *(Lemma 8),*
3. $|\boldsymbol{u}_a^\top \boldsymbol{W}^t \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \boldsymbol{x}^t)(\boldsymbol{v}_a^\top \boldsymbol{y}^t)| > \frac{\epsilon}{4}$ *(Lemma 9(i)),*
4. $\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\mathcal{P}_a(S^t,\boldsymbol{\beta}^t)} \|(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) - (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W})\| > \frac{\delta}{2}$ *(Lemma 10).*

Hence, in iteration t, the algorithm generates a valid inequality (line 21 of Algorithm 2) that is violated by $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$ (this can be seen from parts 1 and 4 of Claim 1). Thus, $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) \notin S^{t+1}$, contradicting $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) \in \tilde{S} \subseteq S^{t+1}$. So, the case that $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) \notin \mathcal{F}^{\frac{\epsilon}{2}}$ will not happen. If $\mathcal{F} \neq \emptyset$, then, this contradiction implies that we must have $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) \in \mathcal{F}^{\frac{\epsilon}{2}}$.

Case 2.2. $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) \in \mathcal{F}^{\frac{\epsilon}{2}}$. Now, consider the case that $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}) \in \mathcal{F}^{\frac{\epsilon}{2}}$.

Claim 2. *There exists a finite $t \in \mathcal{T}$ such that*

1. $(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) \in \Omega^t$ *(Lemma 8),*
2. $|\boldsymbol{u}_a^\top \boldsymbol{W}^t \boldsymbol{v}_a - (\boldsymbol{u}_a^\top \boldsymbol{x}^t)(\boldsymbol{v}_a^\top \boldsymbol{y}^t)| \leq \epsilon$ *for* $a \in [nm]$, *implying* $(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) \in \Omega_\epsilon$ *(Lemma 9(ii)),*
3. $\boldsymbol{f}_0^\top \tilde{\boldsymbol{x}} + \boldsymbol{g}_0^\top \tilde{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \tilde{\boldsymbol{W}} - \underline{z}^t \leq \gamma$ *(using the fact that* $\text{h-lim}_{t\to\infty} S^t = \tilde{S}$ *and similar to the argument in the proof of Lemma 8).*

For a sufficiently large t, $t \in \mathcal{T}$, that satisfies Claim 2, we conclude that $\underline{z}^t \leq \min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\Omega_\epsilon} \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W} \leq \boldsymbol{f}_0^\top \tilde{\boldsymbol{x}} + \boldsymbol{g}_0^\top \tilde{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \tilde{\boldsymbol{W}}$. Note that $\Omega_\epsilon \neq \emptyset$ because $(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{W}^t) \in \Omega_\epsilon$. Also, note that the optimal solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}}) \in \Omega_\epsilon$ to the above optimization problem is such that $\boldsymbol{f}_0^\top \hat{\boldsymbol{x}} + \boldsymbol{g}_0^\top \hat{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \hat{\boldsymbol{W}} - \underline{z}^t \leq \boldsymbol{f}_0^\top \tilde{\boldsymbol{x}} + \boldsymbol{g}_0^\top \tilde{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \tilde{\boldsymbol{W}} - \underline{z}^t \leq \gamma$. If $0 < \gamma \leq \epsilon$, then, the algorithm terminates in the t-th iteration and yields an ϵ-optimal solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$. Otherwise, if $\gamma > \epsilon$, with a similar argument, we can find a sufficiently large t, $t \in \mathcal{T}$, such that $\boldsymbol{f}_0^\top \tilde{\boldsymbol{x}} + \boldsymbol{g}_0^\top \tilde{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \tilde{\boldsymbol{W}} - \underline{z}^t \leq \epsilon$, in addition to the first two parts of the above claim. Note that $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}})$ provides an upper bound to the optimization problem $\min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\in\Omega_\epsilon} \boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W}$. Thus, $\boldsymbol{f}_0^\top \hat{\boldsymbol{x}} + \boldsymbol{g}_0^\top \hat{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \hat{\boldsymbol{W}} - \underline{z}^t \leq \boldsymbol{f}_0^\top \tilde{\boldsymbol{x}} + \boldsymbol{g}_0^\top \tilde{\boldsymbol{y}} + \boldsymbol{A}_0 \bullet \tilde{\boldsymbol{W}} - \underline{z}^t \leq \epsilon$. Consequently, the algorithm terminates in the t-th iteration and yields an ϵ-optimal solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$. In any case, we have an ϵ-optimal solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{W}})$ to the feasible (BLP). □

Remark 2. *It is worth noting that the fact that $\gamma > 0$, and hence the vertex exploration is imposed, is used in part 1 of Claim 1 and in parts 1 and 3 of Claim 2. These parts fail to hold when $\gamma = 0$ as they are proved using a limiting argument.*

4.3 SVD Basis

In Algorithm 2, it is assumed that two sets of bases are available a priori. In this section, we assume that bases are obtained through the course of the algorithm.

Given an optimal solution $(\breve{\boldsymbol{x}}, \breve{\boldsymbol{y}}, \breve{\boldsymbol{W}})$ to the minimization over the relaxation S^t, the left- and right-singular vector $\boldsymbol{u}_t$ and $\boldsymbol{v}_t$, respectively, corresponding to the largest singular value of $\breve{\boldsymbol{W}} - \breve{\boldsymbol{x}}\breve{\boldsymbol{y}}^\top$ are obtained. If $(\boldsymbol{u}_t, \boldsymbol{v}_t)$ is not in the span of previously generated vectors $\mathcal{V}^t$, $(\boldsymbol{u}_t, \boldsymbol{v}_t)$ is added to the set of bases. While the bases are generated, relaxation S^t is also refined by adding disjunctive cuts through procedure `SepCuts`$(\breve{\boldsymbol{x}}, \breve{\boldsymbol{y}}, \breve{\boldsymbol{W}}; S^t, \boldsymbol{u}_t, \boldsymbol{v}_t)$. This procedure of generating bases is continued until two sets of bases for $\mathbb{R}^n$ and $\mathbb{R}^m$ are available, i.e., $|\mathcal{V}_t| \geq \max\{n, m\}$. Because these spaces are finite-dimensional, the procedure of generating bases stops after a finite number of iterations. Once the bases are available, the algorithm continues as in Algorithm 2.

4.4 Discussion

The algorithms in Sections 4.2 and 4.3 need the set of γ-optimal extreme point solutions. Thus, the practical performance of the analyzed algorithms depends on the choice of $\gamma > 0$. A larger choice of γ results in fewer iterations to converge while the computational time per iteration might increase. This is due to exploring a larger set of γ-optimal solutions, solving a relaxation problem with more constraints, and a potentially more demanding cut separation problem. On the other hand, a smaller choice of γ might result in a slower convergence. The choice of γ is problem-dependent and should be tuned to trade off the computational time and improvements obtained from the cuts. In other words, although vertex exploration guarantees finite convergence in theory, not all cuts generated through vertex exploration may have computational benefits.

On a related note, in theory, the set of γ-optimal extreme point solutions can be generated using Simplex pivots of the current extreme point solution. Alternatively, one may first encode basic feasible solutions of the current relaxation using binary variables and obtain a mixed-binary linear program (see the idea in [15]). Now, one can use the solution pool feature of a commercial optimization solver (e.g., CPLEX) to enumerate all near-optimal solutions. Recognizing that this limitation to perform a thorough vertex exploration might incur additional computational burdens, in our numerical experiments in Section 5, we explored a few "promising" near-optimal extreme point solutions using random objective function coefficients.

5 Numerical Experiments

This section presents experiments using a pure cutting plane algorithm analyzed in Section 4.3. The experiments are conducted to investigate the performance of our cuts, particularly relaxation quality. In Section 5.1, we provide general details on the implementation of our algorithm. We report the computational results in Section 5.2.

5.1 Implementation Details

In Section 4, a finitely-convergent algorithm that is based on generating cuts at *all* γ-optimal extreme point solutions of the current relaxation was analyzed. For the computational experiments in Section 5.2, we implemented a more practical version of the algorithm analyzed in Section 4.3, which generates cuts at the current relaxation solution and only *a few* additional near-optimal extreme point solutions. We implemented our analyzed algorithm in C++, using the Eigen 3.3.7 library [11], for the SVD of the residual matrix. With this practical consideration, our algorithm was repeated until a time limit was reached or the algorithm could not find a violated cut.

To obtain valid cuts, we use a procedure referred to as $\texttt{SepCutsCGLP}(\breve{\boldsymbol{x}}, \breve{\boldsymbol{y}}, \breve{\boldsymbol{W}}; S^t, \boldsymbol{u}_t, \boldsymbol{v}_t)$. This procedure proceeds similarly to $\texttt{SepCuts}(\breve{\boldsymbol{x}}, \breve{\boldsymbol{y}}, \breve{\boldsymbol{W}}; S^t, \boldsymbol{u}_t, \boldsymbol{v}_t)$, outlined in Algorithm 1, except for it obtains valid cuts through the CGLP, introduced in Lemma 4. This CGLP is based on a (2×2)-way disjunction, where the disjunction is formed according to $(\boldsymbol{u}_t, \boldsymbol{v}_t)$, corresponding to the largest singular value of $\breve{\boldsymbol{W}} - \breve{\boldsymbol{x}}\breve{\boldsymbol{y}}^\top$. $\texttt{SepCutsCGLP}(\breve{\boldsymbol{x}}, \breve{\boldsymbol{y}}, \breve{\boldsymbol{W}}; S^t, \boldsymbol{u}_t, \boldsymbol{v}_t)$ is different from $\texttt{SepCuts}(\breve{\boldsymbol{x}}, \breve{\boldsymbol{y}}, \breve{\boldsymbol{W}}; S^t, \boldsymbol{u}_t, \boldsymbol{v}_t)$, in the sense that the former utilizes an outer approximation on the convex quadratic terms to form the disjunction. So, it is possible that the minimum distance of a point $(\breve{\boldsymbol{x}}, \breve{\boldsymbol{y}}, \breve{\boldsymbol{W}})$ from the convex hull $\mathcal{P}_{\boldsymbol{c}}(S^t, \boldsymbol{\beta})$, where $\boldsymbol{c} = (\boldsymbol{u}_t, \boldsymbol{v}_t)$, is positive, while the corresponding CGLP cannot find a violated disjunctive cut; hence, our modified algorithm stops. At each iteration of the algorithm, the inequalities fed to the CGLP are given by the McCormick constraints, an outer approximation of the convex quadratic terms, and all the disjunctive cuts added to the relaxation in all the previous iterations. We note for Example 1, a disjunctive cut is obtained by approximating the convex quadratic terms at 2500 equally-spaced points on $[0, 1] \times [0, 2]$, where for all other instances such a cut is obtained by approximating the convex quadratic terms at the current solution.

To obtain a near-optimal solution, we added the constraint $\boldsymbol{f}_0^\top \boldsymbol{x} + \boldsymbol{g}_0^\top \boldsymbol{y} + \boldsymbol{A}_0 \bullet \boldsymbol{W} - \underline{z}^t \leq \gamma$ to the current relaxation S^t and replaced the objective function with a randomly generated objective function. To find this additional point at each iteration, we generated three candidate near-optimal extreme point solutions and chose the point with the highest ℓ_1-norm from the optimal solution to the current relaxation problem. The only exception to the above vertex exploration is our experiments for Example 1, where we used the CGAL 5.5.2 library [35] to enumerate all extreme points. Furthermore, we chose parameter γ to be a percentage of the gap between lb(`MC`) and GUROBI. We provide more details on the vertex exploration and the choice of γ in Section 5.2.

5.2 Computational Results

Our experiments are performed over four sets of instances. First, we consider Examples 1 and 2. Then, we consider a set of randomly generated problems for $n = 10$ and $m \in \{2, 3, 4, 5, 10\}$. For each pair (n, m), we generated five instances with only one bilinear constraint, i.e., $p = 1$ in (1). Finally, we consider the problem instances used in [9], but with some modifications. The instances in [9] are generated for a bilinear optimization problem with an objective function $\boldsymbol{x}^\top \boldsymbol{Q}_0 \boldsymbol{x} + \boldsymbol{y}^\top \boldsymbol{R}_0 \boldsymbol{y} + \boldsymbol{x}^\top \boldsymbol{A}_0 \boldsymbol{y}$, with no bilinear constraint but with box constraints on $\boldsymbol{x}$ and $\boldsymbol{y}$. The dimension of n were set to 20 and 100, with $m \in \{4, 8, 16, 20\}$ and $m \in \{4, 20, 40, 80\}$, respectively. For each pair (n, m), eight instances were generated, where matrix $\boldsymbol{A}_0$ has a density of 50% in half of them and a density of 100% in the other half. Moreover, in each group of four instances where $\boldsymbol{A}_0$ has the same density, the ranks of matrices $\boldsymbol{Q}_0$ and $\boldsymbol{R}_0$ were set to 25, 50, 75, and 100 percent of n and m. For each instance, we changed to the objective function to $\boldsymbol{x}^\top \boldsymbol{Q}_0 \boldsymbol{1} + \boldsymbol{y}^\top \boldsymbol{R}_0 \boldsymbol{1} + \boldsymbol{x}^\top \boldsymbol{A}_0 \boldsymbol{y}$, where $\boldsymbol{1}$ is a vector of ones with an appropriate dimension. In other words, we generated an instance in the form of the objective function in (1), where the i-th entry in vectors $\boldsymbol{f}_0$ and $\boldsymbol{g}_0$ were obtained by the summation of entries in the i-th row of matrix $\boldsymbol{Q}_0$ and $\boldsymbol{R}_0$, respectively.

We solved each problem instance with three pure cutting plane algorithms: (i) the algorithm analyzed in Section 4.3 (denoted as "`Modified SVD(x)`"), where the presence of `x`≥ 1 indicates the number of additional near-optimal extreme point solutions explored for the cut generation and its absence indicates exploring all near-optimal extreme points, (ii) the algorithm studied in [9] as introduced in Section 2.2 (denoted as "`SVD`"), and (iii) the algorithm studied in [6] (denoted as "`BCM`"). Using algorithms `Modified SVD(x)`, `SVD`, and `BCM`, instances were solved with the LP solver CPLEX 12.7. For `Modified SVD(x)` and `SVD`, we set the maximum number of cuts per iteration per extreme point to at most one cut unless otherwise stated. We note that the only difference between algorithms `Modified SVD(x)` and `SVD` is the vertex exploration in the former, and all the other implementation details remained the same. The code for algorithm `BCM` with the LP solver GUROBI is available at `https://github.com/g-munoz/poly_cuts_cpp`. We implemented this algorithm in C++ with the LP solver CPLEX 12.7. For `BCM`, we set the maximum number of cuts added per iteration to 1000 unless otherwise stated. In addition, for comparisons, we solved all instances with the nonconvex option of GUROBI 9.1.2 to obtain a bound, listed under column "GUROBI".

Table 1: Comparison of `BCM` [6], `SVD` [9], and `Modified SVD` on Example 1.

BCM [6]			SVD [9]			Modified SVD				
I-GC (%)	G (%)	# Cuts	I-GC (%)	G (%)	# Cuts	I-GC (%)	G (%)	# Cuts	O-GC BCM (%)	O-GC SVD (%)
96.4723	0.7483	327	99.7095	0.0616	20	99.8896	0.0234	56	96.8701	61.9861

The main goal of our experiments was to compare the relaxation quality of the abovementioned algorithms/solver with a prespecified time limit. For each algorithm $\texttt{alg} \in \{\texttt{Modified SVD(x)}, \texttt{SVD}, \texttt{BCM}\}$, we evaluated the quality of the lower bound, denoted as $\text{lb}(\texttt{alg})$, by calculating the gap as

$$\text{G } \texttt{alg} := \frac{\text{GUROBI} - \text{lb}(\texttt{alg})}{|\text{GUROBI}|} \times 100\%,$$

where GUROBI refers to the bound obtained by GUROBI within the prespecified time limit. Moreover, we calculated the gap closed compared to the initial lower bound obtained from McCormick relaxation, denoted as $\text{lb}(\texttt{MC})$, as

$$\text{I-GC } \texttt{alg} := \frac{\text{lb}(\texttt{alg}) - \text{lb}(\texttt{MC})}{\text{GUROBI} - \text{lb}(\texttt{MC})} \times 100\%.$$

When $\text{GUROBI} = \text{lb}(\texttt{MC})$ (and hence, $\text{lb}(\texttt{alg}) = \text{lb}(\texttt{MC})$ by design), we assume the initial gap closed is 100%. Finally, we computed the gap closed by `Modified SVDx` relative to a benchmark algorithm $\texttt{alg} \in \{\texttt{Modified SVD(x-1)}, \texttt{SVD}, \texttt{BCM}\}$ as

$$\text{O-GC } \texttt{alg} := \frac{\text{lb}(\texttt{Modified SVDx}) - \text{lb}(\texttt{alg})}{\text{GUROBI} - \text{lb}(\texttt{alg})} \times 100\%.$$

In addition, in this section, we report the total time (in seconds) spent to achieve the lower bound ("Time") and the total number of added cuts ("# Cuts"). For `Modified SVDx`, the total time includes the time spent to explore the near-optimal extreme point solutions. For all experiments with `Modified SVD(x)`, we chose parameter γ to be a percentage of the gap between $\text{lb}(\texttt{MC})$ and GUROBI, i.e., $\gamma = \alpha(\text{GUROBI} - \text{lb}(\texttt{MC}))$ for some $\alpha \in (0, 100]$.

All experiments were performed on a Linux Ubuntu 20.04 environment using one single core of a PC with an Intel Core i7-9700 3.00 GHz processor and 32.00 GB of RAM.

5.2.1 Example 1

Recall from Section 2.2 that `SVD` shows a slow rate of convergence for Example 1. We implemented `Modified SVD` by exploring *all* γ-optimal extreme point solutions, where γ was chosen as 0.5% of the gap between $\text{lb}(\texttt{MC})$ and GUROBI. Table 1 shows that `Modified SVD` improved the lower bound over `BCM` and `SVD`. Figure 3 depicts the evolution of the lower bound over the iteration number using `SVD` and `Modified SVD`. `SVD` terminated after 20 iterations and adding 20 cuts, whereas `Modified SVD` terminated after 15 iterations and adding 56 cuts.

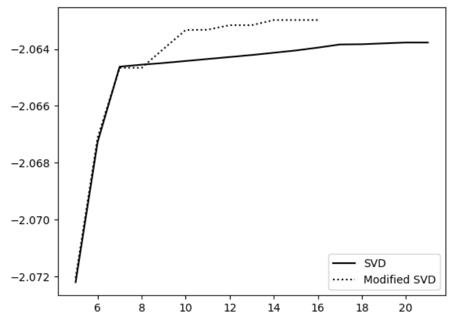


Figure 3: Lower bounds for Example 1 using `SVD` [9] and `Modified SVD`.

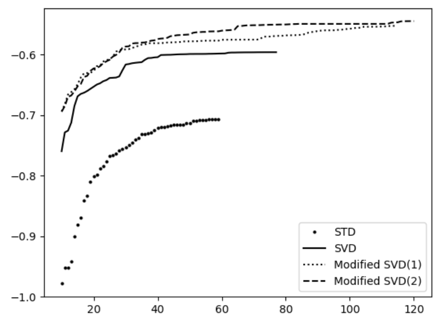


Figure 4: Lower bounds for Example 2 using `STD`, `SVD` [9], `Modified SVD(1)`, and `Modified SVD(2)`.

5.2.2 Example 2

Again, recall from Section 2.2 that `SVD` shows a slow rate of convergence for Example 2. We implemented `Modified SVD(1)` and `Modified SVD(2)` with different values of γ. In particular, $\gamma = \alpha(\mathbf{ub} - \text{lb}(\texttt{MC}))$, with $\alpha \in [0.1, 1]$ in an increment of 0.1% in addition to 5% and 10%. Table 2 shows the result for Example 2. Observe that for all choices of γ, `Modified SVD(2)` improved the lower bound over those of `BCM` and `SVD`. In addition, `Modified SVD(2)` improved the lower bound over those of `Modified SVD(1)` for all choices of γ, except for when $\alpha = 0.8\%$ and $\alpha = 10\%$. On average (over all choices of γ), exploring two near-optimal solutions using `Modified SVD(2)` resulted in a 89.75%, 36.6228%, and 15.6102% reduction in the remaining optimality gap, compared to `BCM`, `SVD`, and `Modified SVD(1)`, respectively. We also note that `Modified SVD(1)` achieved its best lower bound (i.e., lowest optimality gap) when $\alpha = 0.4\%$, whereas `Modified SVD(2)` achieved its best lower bound when $\alpha = 0.7\%$. Table 3 shows a summary of results for all algorithms when `Modified SVD(1)` and `Modified SVD(2)` were implemented with their respective best choice of γ. Figure 4 also depicts the evolution of the lower bound using `STD`, `SVD`, `Modified SVD(1)`, and `Modified SVD(2)` for Example 2. While `SVD` terminated after 76 iterations, `Modified SVD(1)` and `Modified SVD(2)` continued for 113 and 119 iterations, respectively.

Table 2: Comparison of `Modified SVD(1)` and `Modified SVD(2)` on Example 2 with varying γ parameters.

	Modified SVD(1)				Modified SVD(2)						
α	lb	I-GC (%)	G (%)	# Cuts	lb	I-GC (%)	G (%)	# Cuts	O-GC BCM (%)	O-GC SVD (%)	O-GC SVD(1) (%)
0.1	-0.6029	96.5686	20.5885	200	-0.5825	97.2485	16.5088	325	86.0901	13.9932	19.8155
0.2	-0.5821	97.2618	16.4294	236	-0.5574	98.0850	11.4900	261	90.3188	40.1400	30.0647
0.3	-0.5691	97.6971	13.8174	160	-0.5563	98.1222	11.2667	209	90.5070	41.3034	18.4601
0.4	-0.5524	98.2527	10.4841	211	-0.5484	98.3857	9.6861	242	91.8387	49.5378	7.6115
0.5	-0.5572	98.0933	11.4402	212	-0.5453	98.4903	9.0583	240	92.3677	52.8084	20.8205
0.6	-0.5749	97.5028	14.9833	181	-0.5633	97.8896	12.6625	274	89.3309	34.0316	15.4893
0.7	-0.5675	97.7495	13.5032	119	-0.5446	98.5140	8.9159	295	92.4877	53.5504	33.9719
0.8	-0.5614	97.9523	12.2860	182	-0.5692	97.6921	13.8476	176	88.3323	27.8571	-12.7104
0.9	-0.5676	97.7482	13.5107	141	-0.5472	98.4264	9.4416	207	92.0447	50.8115	30.1175
1	-0.5742	97.5258	14.8454	139	-0.5468	98.4395	9.3632	263	92.1108	51.2199	36.9287
5	-0.5865	97.1178	17.2930	138	-0.5849	97.1700	16.9801	203	85.6929	11.5376	1.8091
10	-0.5728	97.5721	14.5672	286	-0.5838	97.2066	16.7604	219	85.8781	12.6826	-15.0555
Average		**97.5868**	**14.4790**			**97.9725**	**12.1651**		**89.7500**	**36.6228**	**15.6102**

Notes: $\gamma = \alpha(\text{GUROBI} - \text{lb}(\texttt{MC}))$.

Table 3: Comparison of `BCM` [6], `SVD` [9], `Modified SVD(1)`, and `Modified SVD(2)` on Example 2.

BCM [6]			SVD [9]			Modified SVD(1)			Modified SVD(2)					
I-GC (%)	G (%)	# Cuts	I-GC (%)	G (%)	# Cuts	I-GC (%)	G (%)	# Cuts	I-GC (%)	G (%)	# Cuts	O-GC BCM (%)	O-GC SVD (%)	O-GC SVD(1) (%)
80.22	118.68	5663	96.8	19.19	76	98.25	10.48	211	98.51	8.92	295	92.49	53.55	14.96

5.2.3 Third Set of Instances

For a set of randomly generated instances, we tested algorithms `BCM`, `SVD`, `Modified SVD(1)`, and `Modified SVD(2)`. We summarize the results in Table 4, averaged over five instances. We note that in the implementation `Modified SVD(1)` and `Modified SVD(2)`, we chose parameter γ to be either 1% or 5% of the gap between lb(`MC`) and GUROBI. The time limit for all instances and algorithm with $m = 2$, 3, 4, 5, or 10, is set to 300, 600, 900, 1200, or 2400 seconds, respectively.

Observe from Table 4 that `SVD` improved the initial gap closed compared to that of `BCM` for all pairs (n, m), on average 83.60% vs. 55.16%. However, the performance of `SVD` is improved by exploring additional near-optimal extreme point solutions. In particular, `Modified SVD(1)` and `Modified SVD(2)` closed 85.98% and 86.20% of the initial gap, respectively. We note that on average, using `Modified SVD(2)` resulted in a 71.29%, 15.20%, and 1.67% reduction in the remaining optimality gap, compared to `BCM`, `SVD`, and `Modified SVD(1)`, respectively. Nevertheless, for instances $(10, 3)$ and $(10, 4)$, we observed that on average, `Modified SVD(1)` yielded a higher-quality solution compared to `Modified SVD(2)`. The improvement over `SVD` was generally achieved at the expense of a higher computational time for `Modified SVD(1)` and `Modified SVD(2)`. As expected, adding more cuts at each iteration increases the size of the relaxation problem, which, in turn, leads to a larger CGLP. So, the overall computational time might increase. On the other hand, while `Modified SVD(1)` and `Modified SVD(2)` yielded a higher-quality solution compared to `BCM`, this improvement was achieved with a comparable computational time. We note that for `BCM`, we also set the maximum number of cuts added per iteration to 20 (similar to the setting in [6]). However, we observed that this new setting resulted in a worse lower bound compared to the results in Table 4; in particular, the initial gap closed reduced from 55.16% to 52.94%, on average.

Table 4: Comparison of BCM [6], SVD [9], Modified SVD(1), and Modified SVD(2) on the second set of instances.

		BCM [6]			SVD [9]			Modified SVD(1)			Modified SVD(2)					
(n, m)	GUROBI	I-GC (%)	G (%)	Time (s)	I-GC (%)	G (%)	Time (s)	I-GC (%)	G (%)	Time (s)	I-GC (%)	G (%)	Time (s)	O-GC BCM (%)	O-GC SVD (%)	O-GC SVD(1) (%)
$(10, 2)$	-0.99	66.01	77.11	202.20	90.94	21.27	10.27	92.15	18.97	129.24	92.59	18.10	155.40	80.60	7.95	14.06
$(10, 3)$	-1.15	59.32	101.93	392.74	89.24	26.94	67.37	91.73	22.17	196.24	91.75	22.30	304.07	79.03	15.76	-2.78
$(10, 4)$	-0.78	50.44	254.54	454.79	86.64	70.40	36.12	90.28	54.00	293.07	88.61	60.64	548.78	76.28	17.22	-15.81
$(10, 5)$	-1.36	55.78	119.48	794.66	79.84	54.29	432.56	81.60	49.88	795.77	83.52	47.11	959.46	65.93	23.77	9.77
$(10, 10)$	-1.54	44.23	111.47	2257.40	71.38	58.92	1259.86	74.14	53.18	1858.40	74.55	53.36	2369.85	54.63	11.33	3.12
Average		**55.16**	**132.91**		**83.61**	**46.36**		**85.98**	**39.64**		**86.20**	**40.30**		**71.29**	**15.21**	**1.67**

5.2.4 Fourth Set of Instances

For this set of instances, we tested algorithms BCM, SVD, and Modified SVD(1). We summarize the results in Table 5, where for each pair (n, m), the average is taken over all instances with the same density of matrix $\boldsymbol{A}_0$. We note that in the implementation of SVD and Modified SVD(1) for this set of instances, we chose parameter γ to be either 1%, 5%, or 10% of the gap between lb(MC) and GUROBI. The time limit for all instances and algorithm with $n = 20$ or 100 is set to 3600 or 7200 seconds, respectively.

Observe from Table 5 that, unlike the second set of instances, where for all instances Modified SVD(1) resulted in a higher-quality solution compared to SVD, we observe a mixed performance for the third set of instances. However, as shown under column "O-GC SVD", Modified SVD(1) is only dominated by a small margin, while there are instances where SVD is dominated by a larger margin. In addition, when the density is 50%, $n = 100$, and $m \in \{20, 40, 80\}$, Modified SVD and Modified SVD(1) could not improve beyond the initial McCormick lower bound; hence, column "O-GC SVD" shows zero for these instances.

When Modified SVD(1) is compared with BCM, we also observe a mixed performance for the third set of instances, unlike the second set of instances where Modified SVD(1) always resulted in a higher-quality solution compared to BCM. In particular, when the density is 100%, BCM generally yielded a higher lower bound compared to Modified SVD(1) when $n = 20$, while Modified SVD(1) always yielded a higher lower bound compared to BCM when $n = 100$. There are two points in order about BCM. We observe an anomaly in the lower bound obtained via BCM for instances $(20, 4)$ and $(20, 8)$ when the density is 50%. For instances $(20, 4)$ with a rank factor 25 and 100 for matrices $\boldsymbol{Q}_0$ and $\boldsymbol{R}_0$, and the instance $(20, 8)$ with a rank factor 50, algorithm BCM resulted in a higher lower bound than the upper bound. In fact, for these instances, the lower bound obtained via the initial McCormick relaxation is equal to the upper bound, and we have a certificate of optimality. We suspect there might be some errors in the implementation of BCM in their accompanying code [6]. For these instances, we assumed the lower bound equals the upper bound and considered the initial gap closed by BCM as 100% in our calculations. Another noteworthy observation is that BCM could not improve the lower bound beyond the initial relaxation for all instances with $n = 100$.

Table 5: Comparison of `BCM` [6], `SVD` [9], `Modified SVD(1)` on the third set of instances, with at most one cut per extreme point for `SVD` and `Modified SVD(1)`.

			`BCM` [6]			`SVD` [9]			`Modified SVD(1)`				
Density (%)	(n, m)	UB	I-GC (%)	G (%)	Time (s)	I-GC (%)	G (%)	Time (s)	I-GC (%)	G (%)	Time (s)	O-GC `BCM` (%)	O-GC `SVD` (%)
50	$(20, 4)^\dagger$	-33697.13	100	0.00	685.41	98.50	0.14	900.11	98.94	0.10	901.51	-1313651.51	7.33
	$(20, 8)^\dagger$	-46463.67	93.14	1.87	3632.11	82.55	4.32	1800.34	82.73	4.24	2701.81	-103.55	0.45
	$(20, 16)$	-88196.47	62.45	14.80	3823.81	46.10	21.36	3602.56	45.97	21.49	3604.42	-44.23	-0.21
	$(20, 20)$	-107278.90	54.99	25.10	3879.06	23.32	42.75	3607.74	18.59	45.32	3612.15	-80.94	-6.28
	$(100, 4)$	-160265.27	0.00	4.04	-*	30.45	1.94	5406.99	32.38	1.83	5405.62	32.38	3.29
	$(100, 20)$	-393434.60	0.00	90.22	-*	0.00	90.22	7219.98	0.00	90.22	7243.43	0.00	0.00
	$(100, 40)$	-795719.00	0.00	82.67	-*	0.00	82.67	7282.11	0.00	82.67	7314.02	0.00	0.00
	$(100, 80)$	-313463.84	0.00	28.24	-*	0.00	28.24	7757.63	0.00	28.24	7728.23	0.00	0.00
100	$(20, 4)$	-38932.61	88.31	5.61	3648.54	89.26	5.12	3602.50	90.40	4.69	3604.77	1.80	47.57
	$(20, 8)$	-87015.21	90.87	5.83	3668.95	76.94	14.09	3602.84	78.31	13.25	3603.67	-391.25	5.95
	$(20, 16)$	-133177.69	80.10	17.17	3844.75	61.04	32.90	3605.06	60.57	33.41	3610.52	-104.68	-1.23
	$(20, 20)$	-147769.05	74.99	26.94	3751.24	53.67	50.48	3606.87	52.19	52.05	3615.02	-92.13	-3.30
	$(100, 4)$	-212377.83	0.00	52.92	-*	50.76	26	7210.10	51.77	25.49	7219.63	51.77	1.98
	$(100, 20)$	-655678.70	0.00	124.99	-*	10.92	111.31	7223.76	9.00	113.72	7278.09	9.00	-2.15
	$(100, 40)$	-1350877.87	0.00	128.41	-*	8.27	117.86	7320.42	7.04	119.41	7348.71	7.04	-1.34
	$(100, 80)$	-593254.64	0.00	31.35	-*	6.96	29.17	7469.35	4.73	29.87	7614.35	4.73	-2.40

Notes: We set the time limit for all algorithms and instances with $n = 20$ and $n = 100$ to 3600 and 7200 seconds, respectively.
† Includes instances that `BCM` resulted in a higher lower bound than the upper bound.
* For these instances, a lower bound could not be found within the time limit.

6 Conclusion

We studied a general nonconvex bilinear program with continuous variables. We analyzed a finitely-convergent disjunctive programming-based pure cutting plane algorithm to obtain a global ϵ-optimal solution of the bilinear program. While the analyzed algorithm partially relies on the ideas investigated [9, 29], we proposed exploring *all* near-optimal extreme point solutions to a current relaxation. We provided a theoretical foundation to demonstrate that generating cuts at all near-optimal solutions guarantees global ϵ-optimality for a bilinear program.

Since exploring all near-optimal extreme point solutions is computationally expensive, we implemented a "practical" version of the analyzed algorithm for our numerical experiments. In fact, we generated valid cuts at the current relaxation solution and only *a few* near-optimal extreme point solutions. The results suggested that this practical implementation improved the optimality gap relative to the optimality gap resulted from the procedure that generates cuts *only* at the optimal extreme point solution using singular value decomposition [9]. Moreover, we also demonstrated the potential of our analyzed algorithm to reduce the optimality gap over to the pure cutting plane approach proposed in [6]. We note that although we conducted a comparative study between our analyzed algorithm and those in [6, 9], our aim in this paper is not to conclude the computational efficiency of one algorithm over another. Instead, our findings should be interpreted as complementary to existing algorithms and inspire incorporating vertex exploration within a branch-and-cut solver. Moreover, we acknowledge that the computational benefits of the analyzed algorithm lie in a judicious vertex exploration to generate disjunctive cuts. Such considerations are problem-dependent and deserve further computational studies and more sophisticated implementations.

Acknowledgement

We thank Jon Lee for providing us with the instances used in [9]. We also thank the authors in [6] for making their codes publicly available.

References

[1] Adams, W. P. and H. D. Sherali (1993). Mixed-integer bilinear programming problems. *Math. Program. 59*(1-3), 279–305.

[2] Al-Khayyal, F. A. and J. E. Falk (1983). Jointly constrained biconvex programming. *Math. Oper. Res. 8*(2), 273–286.

[3] Balas, E. (1998). Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Appl. Math. 89*(1-3), 3–44.

[4] Bao, X., N. V. Sahinidis, and M. Tawarmalani (2011). Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Math. Program. 129*(1), 129.

[5] Barber, C., D. Dobkin, and H. Huhdanpaa (1996). The quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software 22*(4), 469–483. http://www.qhull.org.

[6] Bienstock, D., C. Chen, and G. Munoz (2020). Outer-product-free sets for polynomial optimization and oracle-based cuts. *Math. Program. 183*(1), 105–148.

[7] Davarnia, D., J.-P. P. Richard, and M. Tawarmalani (2017). Simultaneous convexification of bilinear functions over polytopes with application to network interdiction. *SIAM J. Optim. 27*(3), 1801–1833.

[8] Dey, S. S., A. Santana, and Y. Wang (2019). New SOCP relaxation and branching rule for bipartite bilinear programs. *Optim. Eng. 20*(2), 307–336.

[9] Fampa, M. and J. Lee (2021). Convexification of bilinear forms through non-symmetric lifting. *J. Global Optim. 80*(2), 287–305.

[10] Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Sci. 22*(4), 455–460.

[11] Guennebaud, G., B. Jacob, et al. (2010). Eigen v3. http://eigen.tuxfamily.org.

[12] Gupte, A., S. Ahmed, M. S. Cheon, and S. Dey (2013). Solving mixed integer bilinear problems using MILP formulations. *SIAM J. Optim. 23*(2), 721–744.

[13] Harjunkoski, I., T. Westerlund, and R. Pörn (1999). Numerical and environmental considerations on a complex industrial mixed integer non-linear programming (MINLP) problem. *Comput. Chem. Eng. 23*(10), 1545–1561.

[14] Konno, H. (1976). A cutting plane algorithm for solving bilinear programs. *Math. Program. 11*(1), 14–27.

[15] Lee, S., C. Phalakornkule, M. M. Domach, and I. E. Grossmann (2000). Recursive MILP model for finding all the alternate optima in LP models for metabolic networks. *Comput. Chem. Eng. 24*(2-7), 711–716.

[16] Locatelli, M. and U. Raber (2002). Packing equal circles in a square: a deterministic global optimization approach. *Discrete Appl. Math. 122*(1-3), 139–166.

[17] Mathiesen, L. (1985). Computational experience in solving equilibrium models by a sequence of linear complementarity problems. *Oper. Res. 33*(6), 1225–1250.

[18] McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems. *Math. Program. 10*(1), 147–175.

[19] Misener, R. and C. A. Floudas (2009). Advances for the pooling problem: Modeling, global optimization, and computational studies. *Appl. Comput. Math. 8*(1), 3–22.

[20] O'Searcoid, M. (2006). *Metric spaces*. Springer Science & Business Media.

[21] Owen, J. H. and S. Mehrotra (2001). A disjunctive cutting plane procedure for general mixed-integer linear programs. *Math. Program. 89*(3), 437–448.

[22] Pham, V., C. Laird, and M. El-Halwagi (2009). Convex hull discretization approach to the global optimization of pooling problems. *Ind. Eng. Chem. Res. 48*(4), 1973–1979.

[23] Rahimian, H. and S. Mehrotra (2020). Sequential convexification of a bilinear set. Optimization Online http://www.optimization-online.org/DB_HTML/2020/01/7595.html.

[24] Rockafellar, R. T. (1970). *Convex analysis*, Volume 28. Princeton University Press.

[25] Rodríguez, M. A. and A. Vecchietti (2008). Enterprise optimization for solving an assignment and trim-loss non-convex problem. *Comput. Chem. Eng. 32*(11), 2812–2822.

[26] Ruszczyński, A. (2006). *Nonlinear optimization*, Volume 13. Princeton university press.

[27] Salinetti, G. and R. J.-B. Wets (1979). On the convergence of sequences of convex sets in finite dimensions. *SIAM Review 21*(1), 18–33.

[28] Santana, A. and S. S. Dey (2020). The convex hull of a quadratic constraint over a polytope. *SIAM J. Optim. 30*(4), 2983–2997.

[29] Saxena, A., P. Bonami, and J. Lee (2010). Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. *Math. Program. 124*(1-2), 383–411.

[30] Saxena, A., P. Bonami, and J. Lee (2011). Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. *Math. Program. 130*(2), 359–413.

[31] Sherali, H. D. and W. P. Adams (2013). *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*, Volume 31. Springer Science & Business Media.

[32] Sherali, H. D. and A. Alameddine (1992). A new reformulation-linearization technique for bilinear programming problems. *J. Global Optim. 2*(4), 379–410.

[33] Sherali, H. D. and C. Shetty (1980). A finitely convergent algorithm for bilinear programming problems using polar cuts and disjunctive face cuts. *Math. Program. 19*(1), 14–31.

[34] Stubbs, R. A. and S. Mehrotra (1999). A branch-and-cut method for 0-1 mixed convex programming. *Math. Program. 86*(3), 515–532.

[35] The CGAL Project (2023). *CGAL User and Reference Manual* (5.5.2 ed.). CGAL Editorial Board.

[36] Vaish, H. and C. Shetty (1977). A cutting plane algorithm for the bilinear programming problem. *Nav. Res. Logist. Q. 24*(1), 83–94.