Revisiting Degeneracy, Strict Feasibility, Stability, in Linear Programming

Jiyoung Im* Henry Wolkowicz†

Saturday 8th October, 2022

Abstract

Currently, the simplex method and the interior point method are indisputably the most popular algorithms for solving linear programs. Unlike general conic programs, linear programs, **LP**s, with a finite optimal value do not require strict feasibility in order to establish strong duality. Hence strict feasibility is seldom a concern, even though strict feasibility is equivalent to stability and a compact dual optimal set. This lack of concern is also true for other types of degeneracy of basic feasible solutions in **LP**. In this note we discuss that the specific degeneracy that arises from lack of strict feasibility necessarily causes difficulties in both simplex and interior point methods. In particular, we show that the lack of strict feasibility implies that every basic feasible solution, **BFS**, is degenerate; thus conversely, the existence of a nondegenerate **BFS** implies that strict feasibility (regularity) holds. We prove the results using facial reduction and simple linear algebra. In particular, the facially reduced system reveals the implicit non-surjectivity of the linear map of the equality constraint system. As a consequence, we emphasize that facial reduction involves two steps where, the first guarantees strict feasibility, and the second recovers full row rank of the constraint matrix. This illustrates the implicit *singularity* of problems where strict feasibility fails, and also helps in obtaining new efficient techniques for preproccessing. We include an efficient preprocessing method that can be performed as an extension of phase-I of the two-phase simplex method. We show that this can be used to avoid the loss of precision for many classical problems in the literature, e.g., those in the NETLIB problem set.

Keywords: linear programming, facial reduction, preprocessing, degeneracy, implicit problem singularity

AMS Classification: 90C05, 90C49.

Contents

*Department of Combinatorics and Optimization Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1;

†Department of Combinatorics and Optimization Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1; Research supported by The Natural Sciences and Engineering Research Council of Canada; `www.math.uwaterloo.ca/~hwolkowi`.

List of Tables

List of Figures

List of Algorithms

1 Introduction

The *Slater condition* (strict feasibility) is a useful property for optimization models to have. Unlike general conic programs, linear programs (**LP**s) do not require strict feasibility as a constraint qualification to guarantee strong duality, and therefore, it is often not discussed. In fact, degeneracy in general is not considered to be a serious concern in linear programming. The Goldman-Tucker Theorem [28] is related in that it guarantees a primal-dual optimal solution satisfying strict complementarity $x^* + z^* > 0$ for the standard form **LP**. However, it does not guarantee the existence of a strictly feasible primal solution $\hat{x} > 0$. The lack of strict feasibility for an **LP** does not seem to cause problems at first glance, especially when the simplex method is used. In this manuscript, we show that the failure of strict feasibility results in degeneracy problems when simplex-type methods are used. More specifically, the lack of strict feasibility inevitably renders **LP**s degenerate, i.e., *every basic feasible solution is degenerate.*[1] Note that strict feasibility along with full row rank of the linear constraint is the *Mangasarian-Fromovitz* constraint qualification [35]. This is equivalent to a compact dual optimal set and is equivalent to stability with respect to perturbations of the right-hand side.

The simplex method [16] is one of the most popular and successful algorithms for solving linear programs. Degeneracy, a zero basic variable, could result in cycling and noncovergence. There are many anti-cycling rules, see e.g., [7, 17, 26, 47] and the references therein. However, techniques for the resolution of degeneracy often result in *stalling* [6,12,36,43], i.e., result in taking a large number of iterations before leaving a degenerate point. Degeneracies are known to cause numerical issues when interior point methods are used, e.g., [32]. For example, degeneracy can result in singularity of the Jacobian of the optimality conditions, and thus also in ill-posedness and loss of accuracy [30].

[1] Conversely, if we can find *one* nondegenerate basic feasible solution, then strict feasibility holds.

Our main results on the degeneracy arising from loss of strict feasibility are shown using the process called *facial reduction, **FR***. Facial reduction is an effective preprecessing tool to use in the absence of strict feasibility. Given a problem with lack of strict feasibility, facial reduction strives to formulate an equivalent problem so that the reformulation has a Slater point. By examining the facially reduced system, we obtain two results. First, we show that every basic feasible solution is degenerate when strict feasibility fails. This leads to an efficient preprocessing for eliminating variables that are fixed at 0. Second, we understand a source of instability arising in problems that fail strict feasibility. The facially reduced system reveals that the linear map that defines the feasible set is implicitly non-surjective. Finally, we use these results to develope an efficient preprocessing technique to obtain strict feasibility. This technique is illustrated on instances from the NETLIB data set.

1.1 Contributions and Outline

The contribution of this manuscript is threefold.

1. We provide the complete description of the facially reduced system of a linear program.
2. We show that every basic feasible solution of a standard linear program is degenerate when strict feasibility fails.
3. We propose and illustrate an efficient preprocessing scheme that can be performed as an extension of phase-I of the two-phase simplex method. This technique allows for eliminating variables fixed at 0, and thus regularizing and simplifying the **LP**.

The manuscript is organized as follows. In Section 2 we present the background and notations. Included are the notions of degeneracy, facial reduction and three types of singularity degree. We then describe what facial reduction tries to achieve. In Section 3 we present our main result and immediate corollaries. We also present the efficient preprocessing method that can be used as an extension of phase-I of the two-phase simplex method. In addition, we relate our main result to known results in the literature, such as distance to infeasibility. In Section 4 we illustrate algorithmic performance of interior point methods and the simplex method under the lack of strict feasibility. We present our conclusions in Section 5.

2 Preliminaries

2.1 Background and Notation

We let $\mathbb{R}^n, \mathbb{R}^{m\times n}$ be the standard real vector spaces of n-coordinates and m-by-n matrices, respectively. We use $\mathbb{R}^n_+$ ($\mathbb{R}^n_{++}$, resp.) to denote the n-tuple with nonnegative (positive) entries. We use $\langle \cdot,\cdot\rangle$ to denote the usual inner product. Given a vector $x\in\mathbb{R}^n$, we let $\operatorname{supp}(x)$ to denote the index set $\{i : x_i \neq 0\}$. Given a matrix $A\in\mathbb{R}^{m\times n}$, we adopt the MATLAB notation to denote a submatrix of A. Given a subset $\mathcal{I}$ of column indices, $A(:,\mathcal{I})\in\mathbb{R}^{m\times|\mathcal{I}|}$ is the submatrix of A that contains the columns of A in $\mathcal{I}$. We also use the notation $A_\mathcal{I}$ to denote $A(:,\mathcal{I})$ when the meaning is clear. Given a convex set $\mathcal{C}$, $\operatorname{relint}(\mathcal{C})$ denotes the relative interior of the set $\mathcal{C}$.

Throughout this manuscript, we work with feasible **LP**s in standard form with finite optimal value:

$$(\mathcal{P}) \qquad p^* = \min_x \left\{ c^T x \ : \ Ax = b, \ x \geq 0 \right\},$$

where $p^* \in \mathbb{R}, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. We assume that $\operatorname{rank}(A) = m$, i.e., there is no redundant constraint. We use $\mathcal{F}$ to denote the feasible region of $(\mathcal{P})$

$$\mathcal{F} = \{x \in \mathbb{R}^n : Ax = b,\ x \geq 0\}. \tag{2.1}$$

2.1.1 Degeneracy in LP

Given an index set $\mathcal{B} \subset \{1, \ldots, n\}, |\mathcal{B}| = m$, a point $x \in \mathcal{F}$ is called a *basic feasible solution,* ***BFS*** if $A(:, \mathcal{B})$ is nonsingular and $x_i = 0,\ \forall i \in \{1, \ldots, n\} \setminus \mathcal{B}$. It is well-known that the simplex method iterates from **BFS** to **BFS**. A basic feasible solution $x \in \mathcal{F}$ is *nondegenerate* if $x_i > 0,\ \forall i \in \mathcal{B}$; it is *degenerate* if $x_i = 0$, for some $i \in \mathcal{B}$. It is clear that every basic feasible solution has at most m positive entries.[2]

We partition the index set $\{1, \ldots, n\}$ as

$$\{1, \ldots, n\} = \mathcal{I}_+ \cup \mathcal{I}_0, \text{ where } \mathcal{I}_0 := \{i : x_i = 0, \forall x \in \mathcal{F}\} \text{ and } \mathcal{I}_+ = \{1, \ldots, n\} \backslash \mathcal{I}_0,$$

i.e., $\mathcal{I}_0$ denotes the variables *fixed at* 0. Note that fixed variables are identified during preprocessing in the literature if the upper and lower bounds are equal, e.g., [2,33]. However, the set $\mathcal{I}_0$ is not as easily identified. We let $\mathcal{I}_{++} = \{i : x_i > 0, \forall x \in \mathcal{F}\}$ denote the variables that are positive on the feasible set; note $\mathcal{I}_{++} \subseteq \mathcal{I}_+$.

There are in fact several types of degeneracy. Let $\bar{x}$ be a given **BFS** with basis $\mathcal{B}$. (Wlog $\mathcal{B} = \{1, \ldots, m\}$.) We can write the equivalent canonical form representation of the feasible set using the basis at $\bar{x}$:

$$\mathcal{F} = \left\{ x = \begin{pmatrix} x_{\mathcal{B}} \\ x_{\mathcal{N}} \end{pmatrix} : x_{\mathcal{B}} = b - A_{\mathcal{B}}^{-1} A_{\mathcal{N}} x_{\mathcal{N}} \geq 0, x_{\mathcal{N}} \geq 0 \right\}. \tag{2.2}$$

In this form $x_{\mathcal{N}} \in \mathbb{R}^{n-m}_+$, we have n inequality constraints, and we see that degeneracy is equivalent to having an active set with cardinality greater than $n-m$. This divides into two types corresponding to the sets $\mathcal{I}_0, \mathcal{I}_+$, respectively: (i) inequalities that are active in every **BFS** and correspond to variables in $\mathcal{I}_0$ above; (ii) those that are not active in at least one **BFS**. The geometry of (i) is clear as there is no Slater point and $\mathcal{F}$ is a subset of a face of the nonnegative orthant. For (ii) the geometry is that some of the constraints are redundant in one of two ways, i.e., (i) that discarding them does not change the feasible set, or (ii) does not change the optimality conditions if $\bar{x}$ is optimal.

Remark 2.1. *We note that adding redundant constraints is done in e.g., [18, 19] to show that the central path for interior point methods can follow the boundary closely, i.e., behave very poorly. These redundant constraints correspond to a positive variable in each* ***BFS****, i.e., to an inequality in* (2.2) *that is never active. Complementary slackness implies that they correspond to variables fixed at* 0 *in the dual problem, thus emphasizing that* ***FR*** *on the dual could avoid some of these difficulties.*

2.2 Facial Reduction

In this section we describe the concept of facial reduction and present the properties that are used to establish the main result. We emphasize that facial reduction for $(\mathcal{P})$ involves *two* steps: first,

[2] We mainly consider primal degeneracy here, though everything follows through for dual degeneracy. In fact, there are clear connections from complementary slackness between variables positive in every BFS and dual variables fixed at 0.

obtain an equivalent problem with strict feasibility; second, recover full row rank of the constraint matrix. Note that full row rank is *always* lost during the first step.

Let $K \subset \mathbb{R}^n$ be a convex set. A convex set $F \subseteq K$ is called a *face* of K, denoted $F \trianglelefteq K$, if for all $y, z \in K$ with $x = \frac{1}{2}(y+z) \in F$, we have $y, z \in F$. Given a convex set $\mathcal{C} \subseteq K$, the *minimal face* for $\mathcal{C}$ is the intersection of all faces containing the set $\mathcal{C}$.

Proposition 2.2. *[22, Theorem 3.1.3](theorem of the alternative) For the feasible system of (2.1), exactly one of the following statements holds:*

1. *There exists $x \in \mathbb{R}^n_{++}$ with $Ax = b$, i.e., strict feasibility holds;*
2. *There exists $y \in \mathbb{R}^m$ such that*

$$0 \neq z := A^T y \in \mathbb{R}^m_+, \quad \textit{and} \quad \langle b, y \rangle = 0. \tag{2.3}$$

Proposition 2.2 gives rise to a process called *facial reduction.* The *facial reduction,* ***FR***, for an **LP** is a process of identifying the minimal face of $\mathbb{R}^n_+$ containing the feasible set $\mathcal{F} = \{x \in \mathbb{R}^n_+ : Ax = b\}$. By finding the minimal face, we can work with a problem that lies in a smaller dimensional space and that statisfies strict feasibility. The **FR** process, i.e., finding the minimal face, is usually done by solving a sequence of auxiliary systems (2.3). More details on **FR** on general conic problems can be found in [8, 9, 22, 39, 44].

We now describe how the set $\mathcal{F}$ (see (2.1)) is represented after **FR**. Suppose that strict feasibility fails. Then Proposition 2.2 implies that there must exist a nonzero $y \in \mathbb{R}^m$ satisfying

$$\langle x, A^T y \rangle = \langle Ax, y \rangle = \langle b, y \rangle = 0, \ \forall x \in \mathcal{F}. \tag{2.4}$$

Hence, every $x \in \mathcal{F}$ is perpendicular to the nonnegative vector $z = A^T y$. We call this vector $z = A^T y$ an *exposing vector* for $\mathcal{F}$, and let the cardinality of its support be $s_z = |\{i : z_i > 0\}|$. Then $z = \sum_{j=1}^{s_z} z_{t_j} e_{t_j}$, where t_j is in nondecreasing order. We now have

$$0 = \langle z, x \rangle \ \text{ and } \ x, z \in \mathbb{R}^n_+ \implies x_i z_i = 0, \ \forall i,$$

i.e., the positive elements in z identify the corresponding elements in x that are *fixed at* 0. Then $x = \sum_{j=1}^{n-s_z} x_{s_j} e_{s_j}$, where s_j is in nondecreasing order. We define the matrix with unit vectors for columns

$$V = \begin{bmatrix} e_{s_1} & e_{s_2} & \dots & e_{s_{n-s_z}} \end{bmatrix} \in \mathbb{R}^{n \times (n-s_z)}.$$

Then we have

$$\mathcal{F} = \{x \in \mathbb{R}^n_+ : Ax = b\} = \{x = Vv \in \mathbb{R}^n : AVv = b, v \in \mathbb{R}^{n-s_z}_+\}. \tag{2.5}$$

We call this matrix $V \in \mathbb{R}^{n \times (n-s_z)}$ a *facial range vector.* The facial range vector restricts the support of all feasible x. We use the identification (2.5) throughout this manuscript. This concludes the first step of **FR**, i.e., guaranteeing the strict feasibility.

It is known that every facial reduction step results in at least one constraint being redundant, see e.g., [9], [34, Lemma 2.7], and [44, Section 3.5]. For completeness we now include a short proof tailored to **LP**, see Lemma 2.3.

Lemma 2.3. *Consider the facially reduced feasible set*

$$\mathcal{F}_r = \left\{ v : AVv = b, v \in \mathbb{R}^{n-s_z}_+ \right\}.$$

Then at least one linear constraint of the ***LP*** *is redundant.*

Proof. Let $z = A^T y$ be the exposing vector satisfying the auxiliary system (2.3). And let V be a facial range vector induced by z. Then

$$0 = V^T z = V^T A^T y = (AV)^T y = \sum_{i=1}^{m} y_i((AV)^T)_i. \tag{2.6}$$

Since $y \in \mathbb{R}^m$ is a nonzero vector, the rows of AV are linearly dependent. □

We now see the result of the full two step facial reduction process, i.e., we get a constraint matrix of full row rank:

$$\mathcal{F} = \{x \in \mathbb{R}^n_+ : Ax = b\} = \{x = Vv \in \mathbb{R}^n : P_{\bar{m}} AVv = P_{\bar{m}} b,\ v \in \mathbb{R}^{n-s_z}_+\}, \tag{2.7}$$

where $P_{\bar{m}} : \mathbb{R}^m \to \mathbb{R}^{\bar{m}}$, $\bar{m} = \text{rank}(AV)$, is the simple projection that chooses the linearly independent rows of AV. We emphasize the importance of this projection by relating it to the so-called *distance to infeasibility* in Section 3.3.1 below. This concludes the second step of **FR**, i.e., guaranteeing the full rank.

For a general conic problem, such as semidefinite programs (**SDP**), the facial reduction iterations do not necessarily end in one iteration; see [14, 44, 45]. And there is a special name for the minimum length of **FR** iterations.

Definition 2.4 ([46, Sect. 4]). *Given a spectrahehedron $\mathcal{S}$, the singularity degree,* $\text{SD}(\mathcal{S})$ *of $\mathcal{S}$ is the* <u>*smallest*</u> *number of facial reduction iterations for finding* $\text{face}(\mathcal{S})$*.*

However, for **LP**s, it is known that **FR** can be done in *one* iteration, i.e., $\text{SD}(\mathcal{F}) \le 1$; see [22, Theorem 4.4.1]. This is due to the fact that the image $A(\mathbb{R}^n_+)$ is polyhedral, and hence the image is facially exposed. In contrast to **FR** performed on conic programs such as the class of **SDP**s, **FR** performed on **LP**s does not alter the sparsity pattern of the data matrix A other than deleting certain columns and rows of A. Moreover, for the **LP** case with a Slater point, $|\mathcal{I}_0| = 0$, we have $\dim(\mathcal{F}) = n - m$, i.e., it is the same as the linear manifold determined by $Ax = b$. In general, by Lemma 2.3 and $\text{SD}(\mathcal{F}) \le 1$, we get

$$\dim(\mathcal{F}) \le n - |\mathcal{I}_0| - (m - \text{SD}(\mathcal{F})).$$

This gives rise to the following novel modified definition.

Definition 2.5. *Let $K \subseteq \mathbb{R}^n$ be a closed convex cone with corresponding feasible set $\mathcal{S} = \{x \in K : Ax = b\}$ and facially reduced feasible set $\{v \in PK : (PA\mathcal{V})(v) = Pb, v \in \mathbb{R}^r\}$, where $PA\mathcal{V}$ is onto $\mathbb{R}^{m_r}$ and PK is the cone defined over the smaller dimensional space. Then the implicit problem singularity,* $\text{IPS}(\mathcal{S}) = m - m_r$*.*
Moreover, the max-singularity degree of $\mathcal{S}$, denoted $\text{maxSD}(\mathcal{S})$*, is the* <u>*largest*</u> *number of nontrivial facial reduction iterations for finding* $\text{face}(\mathcal{S})$*.*

The singularity degree is used in [46, Sect. 4] for providing a Hölder regularity constant for semidefinite programs. This is then used in [21] to derive a convergence rate for alternating projection methods for **SDP**. Note that $\text{maxSD}(\mathcal{S})$ can be a larger lower bound of $\text{IPS}(\mathcal{S})$ than $\text{SD}(\mathcal{S})$, since at least one linear constraint becomes redundant at each **FR** iteration. The effect on ill-conditioning of larger values of IPS is seen in Section 4.1.5.[3]

[3] Definition 2.5 can be used to strengthen the upper bound on the rank of **SDP** solutions in [34], i.e., we get $t(r) \le m - \text{IPS}(\mathcal{S}) \le m - \text{maxSD}(\mathcal{S}) \le m - \text{SD}(\mathcal{S}) \le m$, where $t(r)$ is the triangular number of the rank r.

2.2.1 Preprocessing in LP

An essential step for simplex and interior point methods is preprocessing, see e.g., [2, 29, 33] and the references therein. One specific preprocessing step refers to detecting a *fixed variable*. These are generally detected when the upper and lower bounds on a variable are equal. Fixed variables can also be detected when an invertible block A_{11} can be isolated $A = \begin{bmatrix} A_{11} & A_{12} = 0 \\ A_{21} & A_{22} \end{bmatrix}$, $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$. With $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, we can eliminate $x_1 = A_{11}^{-1} b_1$ and discard the first block of now redundant rows, along with the first block of columns. If $b_1 = 0$ then we have trivially identified variables fixed at zero and removed redundant rows and columns. The remaining block A_{22} remains full row rank as happens in Gaussian elimination.

In general, **FR** for linear programs refers to identifying variables fixed at 0, and removing them along with corresponding columns and redundant rows. In general, this is not as simple as above, and the theorem of the alternative is needed. As a consequence of our main result, we see below that a single step of the simplex method, a phase-I part B approach, yields many of these variables that are identically zero on the feasible set.

One of the standard assumptions in linear programming is full row rank of A. As we observed in Lemma 2.3, each **FR** step results in linear dependence of the constraints. We now summarize two available methods for extracting a maximal linearly independent subset of rows of AV. The first method uses a rank-revealing QR decomposition[4]. Let $M = (AV)^T$. Let $MI(:, \pi) = QR$ be a QR factorization where π is a permutation vector, Q is a orthogonal matrix and R is an upper triangular matrix with a non-increasing diagonal. The matrix $I(:, \pi)$ permutes the columns of M. If M has linearly dependent columns, then the matrix R contains zeros on its diagonal. Let r be the nonzero diagonal entries of R. Then, $\pi(1 : r)$ returns the subset of columns indices of M that are linearly independent. Another available method makes use of artificial variables [15, Box 8.2]. It constructs $\begin{bmatrix} I & AV \end{bmatrix}$ and sets the initial basis matrix to be the first m columns. Then it performs a variant of the phase-I of the two-phase simplex method to drive the basic variables out of the basis one by one. When such an operation is not applicable, a linearly dependent row of AV is detected. Computational improvements of this method are made in [1, 37].

3 Main Result and Consequences

In this section we present our main result, see Theorem 3.1. We provide two proofs: one takes an algebraic approach by using the definition of the basic feasible solution; and the other takes a geometric approach by using extreme points. Both proofs rely heavily on Lemma 2.3. In Section 3.2 we present an efficient preprocessing scheme that can be used as an extension of the phase-I of the two-phase simplex method. In Section 3.3 we include immediate corollaries of the main result and interesting discussions.

3.1 Lack of Strict Feasibility and Relations to Degeneracy

Theorem 3.1. *Suppose that strict feasibility fails for $\mathcal{F}$. Then every basic feasible solution to $\mathcal{F}$ is degenerate.*

[4] https://www.mathworks.com/matlabcentral/fileexchange/77437

3.1.1 An Algebraic Proof of Theorem 3.1 via the Definition of BFS

Proof. Since there is no strictly feasible point in $\mathcal{F}$, there exists a facial range vector V, and as in (2.5) we have

$$\mathcal{F} = \{x \in \mathbb{R}^n \ : \ AVv = b, \ v \in \mathbb{R}^{n-s_z}_+\}.$$

By Lemma 2.3, AV has at least one redundant row. By permuting the columns of A, we may assume that the matrix V is of the form

$$V = \begin{bmatrix} I_r \\ 0 \end{bmatrix} \text{ and } r = n - s_z.$$

We partition the index set $\{1, \ldots, n\}$ as

$$\{1, \ldots, n\} = \mathcal{I}_+ \cup \mathcal{I}_0, \text{ where } \mathcal{I}_+ = \{1, \ldots, r\} \text{ and } \mathcal{I}_0 = \{r+1, \ldots, n\}.$$

Then we have $A = \begin{bmatrix} A(:, \mathcal{I}_+) & A(:, \mathcal{I}_0) \end{bmatrix}$. Let $\bar{x} \in \mathcal{F}$ be a basic feasible solution with basic indices

$$\mathcal{B} \subset \{1, \ldots, n\}, \ |\mathcal{B}| = m, \ \det(A(:, \mathcal{B})) \neq 0, \quad \text{and} \quad A(:, \mathcal{B})\bar{x}(\mathcal{B}) = b.$$

Suppose $\mathcal{B} \subseteq \mathcal{I}_+$. We note, by Lemma 2.3 again, that $A(:, \mathcal{I}_+) = AV$ has redundant rows, i.e., $\operatorname{rank}(A(:, \mathcal{I}_+)) < m$. Hence $\bar{x}$ must include a basic variable in $\mathcal{I}_0$ and this concludes that every basic feasible solution is degenerate. □

3.1.2 A Geometric Proof Using Extreme Points

We now give the second proof of our main result. We first employ the statement presented in [38]. In Proposition 3.2 below, $\mathbb{S}^n_+$ denotes the set of n-by-n positive semidefinite matrices.

Proposition 3.2. *[38, Theorem 2.1] Suppose that $X \in F$, where F is a face of the set $\{X \in \mathbb{S}^n_+ : \operatorname{trace}(A_i X) = b_i, \forall i = 1, \ldots, m\}$. Let $d = \dim F$, $r = \operatorname{rank}(X)$. Then $\frac{r(r+1)}{2} \leq m + d$.*

The set in Proposition 3.2 is called a spectrahedron. Feasible sets of standard semidefinite programs are represented as spectrahedra. A sepectrahedron is a generalization of the polyhedral set $\mathcal{F}$ and the proof from [38, Theorem 2.1] can be altered to work with $\mathcal{F}$. We include the proof for completeness.

Corollary 3.3. *Suppose that $x \in F$, where F is a face of the set $\mathcal{F}$. Let r be the number of nonzeros in x and $d = \dim F$. Then the number of nonzero entries of $x \in F$ is at most $m + d$.*

Proof. Let $x \in F$ and let r be the number positive entries in x. Let $\bar{x} \in \mathbb{R}^r$ be the vector obtained by discarding the 0 entries in x. This is readily given by the following matrix-vector multiplication $\bar{x} = I(\operatorname{supp}(x), :)x$, where $\operatorname{supp}(x)$ is the support of x, the set of indices $\{i : x_i > 0\}$. Let $\bar{A} \in \mathbb{R}^{m \times r}$ be the matrix after removing the columns of A that are not in the support of x, i.e., $\bar{A} = A(:, \operatorname{supp}(x))$. We note that $\bar{x}$ is a particular solution to the system $\bar{A}z = b$ and $\bar{x} > 0$.

Suppose to the contrary that $r > m + d$. Since $r - m > d$, there exists at least $d + 1$ linearly independent vectors, say $v_1, \ldots, v_{d+1} \in \mathbb{R}^r$, satisfying $\bar{A}v_i = 0, \ \forall i = 1, \ldots, d+1$. For each $i \in \{1, \ldots, d+1\}$ and for $\epsilon \in \mathbb{R}$, we define

$$\begin{array}{ll} v_{i,+} := \bar{x} + \epsilon v_i, & v_{i,-} := \bar{x} - \epsilon v_i, \\ x_{i,+} := I(:, \operatorname{supp}(x))\left(\bar{x} + \epsilon v_i\right), & x_{i,-} := I(:, \operatorname{supp}(x))\left(\bar{x} - \epsilon v_i\right). \end{array}$$

For a sufficiently small ϵ, we have $x_{i,+}, x_{i,-} \in \mathcal{F}$. We note that $x = \frac{1}{2}(x_{i,+} + x_{i,-}),\ \forall i$. Hence, by the definition of face, $x_{i,+} \in F,\ \forall i$. Therefore, F contains vectors $\{x_{i,+}\}_{i=1,\ldots,d+1} \cup \{x\}$ that are affinely independent and hence $\dim(F) \geq d+1$. □

A point x in a convex set $\mathcal{C}$ is called an *extreme point* if, for all $y, z \in \mathcal{C}$, $x = \frac{1}{2}(y+z)$ implies $x = y = z$. An extreme point is itself a face and the dimension of this face is 0. Hence, we obtain Corollary 3.4 by writing Corollary 3.3 through the lens of extreme points.

Corollary 3.4. *Every extreme point $x \in \mathcal{F}$ has at most m positive entries.*

We now restate the main result of this paper Theorem 3.1 in the language of extreme points and number of rows of A.

Theorem 3.5. *Suppose that strict feasibility of $\mathcal{F}$ fails. Then every extreme point $x \in \mathcal{F}$ has at most $m-1$ positive entries.*

Proof. Since strict feasibility fails for $\mathcal{F}$, we have $\mathcal{F} = \{x = Vv \in \mathbb{R}^n : AVv = b,\ v \in \mathbb{R}^{n-s_z}_+\}$; see (2.5). From Lemma 2.3, we note that at least one equality in $AVv = b$ is redundant. Let $P_{\bar{m}}AVv = P_{\bar{m}}b$ be the system obtained after removing redundant rows of AV; see (2.7). Then, by Corollary 3.4, every extreme point of the set $\{v \in \mathbb{R}^{n-s_z}_+ : P_{\bar{m}}AVv = P_{\bar{m}}b\}$ has at most $m-1$ nonzero entries. Hence, the statement follows. □

Remark 3.6. *The idea used in the proof of Theorem 3.5 is the same as the one presented in [34] for a spectrahedron. In [34], the authors use Proposition 3.2 to strengthen the bound called the Barvinok-Pataki bound. The bound is strengthened by the means of singularity degree that stems from the facial reduction algorithm [22, 39, 44]. The number of nonzeros in x in Theorem 3.5 plays the role of* $\operatorname{rank}(X)$ *in Proposition 3.2. Facial reduction applied to spectrahedra also yields redundant constraints and hence a similar result follows for spectrahedra.*

3.1.3 Immediate Consequences of Main Result

We first note that Theorem 3.1 and Theorem 3.5 are equivalent owing to the well-known characterization:

$$x \in \mathcal{F} \text{ is a basic feasible solution} \iff x \in \mathcal{F} \text{ is an extreme point.}$$

We now highlight that Theorem 3.1 and Theorem 3.5 do not merely imply the existence of a *single* degenerate basic feasible solution; but rather that *every* basic feasible solution is degenerate. Developing a pivot rule that prevents the simplex method from visiting degenerate points is not possible as it can never avoid degeneracies when strict feasibility fails, as we now illustrate in the following.

Example 3.7. *Consider $\mathcal{F}$ with the data*

$$A = \begin{bmatrix} 1 & 1 & 3 & 5 & 2 \\ 0 & 1 & 2 & -2 & 2 \end{bmatrix} \text{ and } b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Consider the vector $y = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$. *Then*

$$A^T y = \begin{pmatrix} 1 & 0 & 1 & 7 & 0 \end{pmatrix}^T \text{ and } b^T y = 0.$$

*Hence, Proposition 2.2 certifies that $\mathcal{F}$ does not contain a strictly feasible point. There are exactly six feasible bases in $\mathcal{F}$. The **BFS** associated with $\mathcal{B} \in \{\{1,2\},\{2,3\},\{2,4\}\}$ is $x = \begin{pmatrix}0 & 1 & 0 & 0 & 0\end{pmatrix}^T$; and the **BFS** associated with $\mathcal{B} \in \{\{1,5\},\{3,5\},\{4,5\}\}$ is $x = \begin{pmatrix}0 & 0 & 0 & 0 & \frac{1}{2}\end{pmatrix}^T$. Clearly, all basic feasible solutions are degenerate.*

Recall that strict feasibility is equivalent to the Mangasarian-Fromovitz constraint qualification, [40]. The latter is equivalent to stability with respect to perturbations of b, and to a compact dual optimal set. Therefore, the following Corollary 3.8, obtained by writing the contrapositive of Theorem 3.1, is extremely interesting and important. We provide Example 3.9 below to illustrate Corollary 3.8.

Corollary 3.8. *Suppose that there exists a nondegenerate basic feasible solution. Then there exists a strictly feasible point $0 < \hat{x} \in \mathcal{F}$.*

Example 3.9. *Consider $\mathcal{F}$ with the data*

$$A = \begin{bmatrix}1 & 0 & -2 & 3 & -4\\ 0 & -1 & -2 & 3 & 1\end{bmatrix} \text{ and } b = \begin{pmatrix}1\\1\end{pmatrix}.$$

*The system $\mathcal{F}$ has exactly four feasible bases; the **BFS** associated with $\mathcal{B} \in \{\{1,4\},\{2,4\},\{4,5\}\}$ is $x = \begin{pmatrix}0 & 0 & 0 & 1/3 & 0\end{pmatrix}^T$ and the **BFS** associated with $\mathcal{B} = \{1,5\}$ is $x = \begin{pmatrix}5 & 0 & 0 & 0 & 1\end{pmatrix}^T$. We note that the **BFS** associated with $\mathcal{B} = \{1,5\}$ is nondegenerate. As Corollary 3.8 states, the system $\mathcal{F}$ has a strictly feasible point, and it is verified by the point $\frac{1}{10}\begin{pmatrix}4 & 1 & 1 & 4 & 1\end{pmatrix}^T$.*

Corollary 3.8 provides a useful check for strict feasibility when the simplex method is used, i.e., if there is any simplex iteration that yields a nondegenerate **BFS**, then it is useful to record that occurrence. We emphasize that recording the occurrence of a nondegenerate iteration is inexpensive and the occurrence gives a certificate of the stability of the **LP** instance. We revisit Corollary 3.8 in Section 3.2.1 below and present an efficient algorithm for obtaining a Slater point from a nongenerate basic feasible solution. But, Example 3.10 below shows that the converse of Theorem 3.1 and Theorem 3.5 is not true. In other words, strict feasibility holds and every basic feasible solution is degenerate.

Example 3.10.

1. *Consider $\mathcal{F}$ with the data*

$$A = \begin{bmatrix}1 & 0 & 2 & 0 & -2\\ 1 & -3 & 2 & 1 & -2\end{bmatrix} \text{ and } b = \begin{pmatrix}1\\1\end{pmatrix}.$$

   $\mathcal{F}$ has exactly four feasible bases and all of them are degenerate; the basic feasible solution associated with $\mathcal{B} \in \{\{1,2\},\{1,4\}\}$ is $x = \begin{pmatrix}1 & 0 & 0 & 0 & 0\end{pmatrix}^T$ and the basic feasible solution associated with $\mathcal{B} \in \{\{2,3\},\{3,4\}\}$ is $x = \begin{pmatrix}0 & 0 & 1/2 & 0 & 0\end{pmatrix}^T$. However, $\mathcal{F}$ contains a strictly feasible point $\frac{1}{10}\begin{pmatrix}1 & 1 & 5.5 & 3 & 1\end{pmatrix}^T$.

2. *Note that the linear assignment problem (marriage problem) has a strictly feasible point but all the **BFS** are highly degenerate[5]. Therefore, $\mathcal{I}_0 = \emptyset$; the set of variables fixed at 0 is empty. Moreover, as an **LP**, the problem is stable with respect to perturbations in the data.*

[5]Note that this is true for the transportation and the assignment problems. Both are highly degenerate at each **BFS** but satisfy strict feasibility. For example, for the assignment problem order n, the feasible set can be considered to be the doubly stochastic matrices X. The extreme points are the permutation matrices by the Birkoff-Von Neumann theorem. Therefore, each extreme point has exactly n positive elements while there are $m = 2n-1$ linearly independent constraints.

Given a **BFS** $\bar{x} \in \mathcal{F}$, we let the *degree of degeneracy* of $\bar{x}$ denote the number of 0's among its basic variables. By exploiting the facially reduced model we can check how degenerate the **BFS**s of $\mathcal{F}$ are.

Corollary 3.11. *Suppose that strict feasibility fails for $\mathcal{F}$, and let $\mathcal{F}$ have the facial range vector representation in (2.5). Recall that the set of indices $\mathcal{I}_0 = \{i \in \{1, \ldots, n\} : x_i = 0, \ \forall x \in \mathcal{F}\}$. Let $\bar{x} \in \mathcal{F}$ be a basic feasible solution with basis $\mathcal{B}$. Then, the following holds.*

1. *The basis $\mathcal{B}$ has an nonempty intersection with $\mathcal{I}_0$, i.e., $\mathcal{B} \cap \mathcal{I}_0 \neq \emptyset$.*
2. *If the degree of degeneracy of $\bar{x}$ is exactly one, with $\bar{x}_k = 0, k \in \mathcal{B}$, then $x_k, A_{:,k}$ can be discarded from the problem.*
3. *The degree of degeneracy of $\bar{x}$ is at least $m - \operatorname{rank}(AV)$.*
4. *At least $m - \operatorname{rank}(AV)$ number of basic indices of $\bar{x}$ are contained in $\mathcal{I}_0$.*

Proof.

1. Let $\bar{x} \in \mathcal{F}$ be a basic feasible solution and let $\mathcal{B}$ be a basis for $\bar{x}$. Item 1 follows from the proof and the definition of the set $\mathcal{I}_0$ of elements x_i that are identically zero on the feasible set.
2. The proof follows from the algebraic proof of Theorem 3.1 given in Section 3.1.1. Since every **BFS** is degenerate and the basis has a nonempty intersection with $\mathcal{I}_0$, the index k must be in $\mathcal{I}_0$.
3. For Item 3, we note that $A(:, \mathcal{B})$ contains linearly independent columns. Then $A(:, \mathcal{B})$ can contain at most $\operatorname{rank}(AV)$ number of columns from AV. Thus, $\bar{x}(\mathcal{B})$ must contain at least $m - \operatorname{rank}(AV)$ number of zeros.
4. Item 4 is a direct consequence of Item 1 and Item 3.

□

Items 3 and 4 of Corollary 3.11 are closely related to the implicit problem singularity, IPS, and the max-singularity degree, maxSD; see Definition 2.5.

We conclude the discussions with the following interesting observation. This again illustrates the implicit singularity of the constraints when the Slater condition fails.

Corollary 3.12. *Suppose that strict feasibility fails for $\mathcal{F}$ and that $m = 1$. Then the trivial $x^* = 0$ is an optimal solution.*

3.2 Efficient Preprocessing for Facial Reduction and Strict Feasibility

In this section we present an efficient preprocessing method for obtaining a facially reduced system. In Section 3.2.1 we discuss obtaining a strictly feasible point using a nondegeneate **BFS** and its variant. In Section 3.2.2 we consider the general case of finding an exposing vector to obtain the facially reduced strictly feasible **LP**.

3.2.1 Towards a Strictly Feasible Point from a Nondegenerate BFS

By Corollary 3.8, the existence[6] of a nondegenerate **BFS** guarantees the existence of a strictly feasible point. We now propose a process for acquiring a Slater point from a nondegenerate **BFS**, and include a generalization. The arguments in this section also provide a constructive proof of Corollary 3.8.

Let $\bar{x} \in \mathcal{F}$ be a nondegenerate **BFS**. Without loss of generality, we assume that the (all positive) basic variables $\bar{x}_{\mathcal{B}}$ of $\bar{x}$ are located at the last m entries of $\bar{x}$. We fix a scalar $\hat{\gamma} \in (0,1)$ and an index $j \in \{1,\dots,n-m\}$. For some $\alpha \geq 0$, we consider the simplex method *ratio test* type inequality

$$\hat{\gamma}\bar{x}_{\mathcal{B}} - \alpha(A_{\mathcal{B}})^{-1}A_j \geq 0. \tag{3.1}$$

Since $\bar{x}_{\mathcal{B}} > 0, \hat{\gamma} > 0$, there exists a *positive* α that maintains the inequality (3.1). Let

$$\alpha^* = \min\left\{1,\ \max\{\alpha \in \mathbb{R}_+ : \hat{\gamma}\bar{x}_{\mathcal{B}} - \alpha(A_{\mathcal{B}})^{-1}A_j \geq 0\}\right\}, \tag{3.2}$$

and decompose

$$\hat{\gamma}\bar{x}_{\mathcal{B}} = \left(\hat{\gamma}\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j\right) + \alpha^*(A_{\mathcal{B}})^{-1}A_j.$$

We observe that

$$\begin{array}{rcl} b &=& A_{\mathcal{B}}\bar{x}_{\mathcal{B}} \\ &=& (1-\hat{\gamma})A_{\mathcal{B}}\bar{x}_{\mathcal{B}} + \hat{\gamma}A_{\mathcal{B}}\bar{x}_{\mathcal{B}} \\ &=& (1-\hat{\gamma})A_{\mathcal{B}}\bar{x}_{\mathcal{B}} + A_{\mathcal{B}}\left(\hat{\gamma}\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j + \alpha^*(A_{\mathcal{B}})^{-1}A_j\right) \\ &=& A_{\mathcal{B}}(\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j) + \alpha^*A_j. \end{array}$$

If we set $x_j = \alpha^* > 0$ and replace $\bar{x}_{\mathcal{B}}$ by $\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j$, then we have increased the cardinality of the positive entries of a solution. We note that $\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j$ only has strictly positive entries since it it a sum of a positive vector and a nonnegative vector;

$$\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j = \underbrace{(1-\hat{\gamma})\bar{x}_{\mathcal{B}}}_{\text{positive}} + \underbrace{\hat{\gamma}\bar{x}_{\mathcal{B}} - \alpha^*(A_{\mathcal{B}})^{-1}A_j}_{\text{nonnegative}}.$$

We can continue to increase the number of positive entries of a solution one by one for each $j \in \{1,\dots,n-m\}$. Moreover, we can achieve this by a compact vectorized operation. The main idea is that we can choose $\hat{\gamma}$ in (3.1) independently for each $j \in \{1,\dots,n-m\}$. Let γ_j be a positive real number such that $0 < \gamma := \sum_{j=1}^{n-m}\gamma_j < 1$. Then, we have

$$\bar{x}_{\mathcal{B}} = (1-\gamma)\bar{x}_{\mathcal{B}} + \gamma\bar{x}_{\mathcal{B}} = (1-\gamma)\bar{x}_{\mathcal{B}} + \sum_{j=1}^{n-m}\gamma_j\bar{x}_{\mathcal{B}}.$$

We set an auxiliary matrix

$$\Theta = \begin{bmatrix}\gamma_1\bar{x}_{\mathcal{B}} & \cdots & \gamma_{n-m}\bar{x}_{\mathcal{B}}\end{bmatrix} - (A_{\mathcal{B}})^{-1}A(:,1:n-m) \in \mathbb{R}^{m\times(n-m)}$$

and perform (3.2) on each column j of Θ to obtain the vector θ^*:

$$\theta_j^* := \begin{cases}\max(\Theta(:,j)) & \text{if } \max(\Theta(:,j)) \leq 1, \\ 1 & \text{otherwise.}\end{cases}$$

[6] Determining the existence of a degenerate basic feasible solution is an NP-complete problem; see [11].

Then the point

$$\begin{bmatrix} \theta^* \\ \bar{x}_{\mathcal{B}} - (A_{\mathcal{B}})^{-1} A(:, 1:n-m)\theta^* \end{bmatrix}$$

is a strictly feasible point to $\mathcal{F}$. Hence, this operation provides a constructive proof of Corollary 3.8.

We now extend the aforementioned procedure for obtaining a strictly feasible point using any feasible solution $\bar{x} \in \mathcal{F}$ such that $A(:, \mathrm{supp}(\bar{x}))$ is full row rank. We partition $\bar{x} \in \mathcal{F}$ as follows

$$\bar{x} = \begin{pmatrix} \bar{x}_{\mathcal{B}_1} \\ \bar{x}_{\mathcal{B}_2} \\ \bar{x}_{\mathcal{N}} \end{pmatrix}, \text{ where } \mathrm{supp}(\bar{x}) = \mathcal{B}_1 \cup \mathcal{B}_2,\ \mathrm{rank}(A(:, \mathcal{B}_1)) = m, \text{ and } \mathcal{N} = \{1, \ldots, n\} \backslash \mathrm{supp}(\bar{x}). \tag{3.3}$$

We partition A using the same partition $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{N}$:

$$\begin{bmatrix} A_{\mathcal{B}_1} & A_{\mathcal{B}_2} & A_{\mathcal{N}} \end{bmatrix} \bar{x} = b \iff \begin{bmatrix} A_{\mathcal{B}_1} & A_{\mathcal{N}} \end{bmatrix} \begin{pmatrix} \bar{x}_{\mathcal{B}_1} \\ \bar{x}_{\mathcal{N}} \end{pmatrix} = \bar{b} := b - A_{\mathcal{B}_2} x_{\mathcal{B}_2}.$$

Then we can apply the aforementioned procedure to the system

$$\begin{bmatrix} A_{\mathcal{B}_1} & A_{\mathcal{N}} \end{bmatrix} \begin{pmatrix} \bar{x}_{\mathcal{B}_1} \\ \bar{x}_{\mathcal{N}} \end{pmatrix} = \bar{b}$$

and distribute positive weights to $\bar{x}_{\mathcal{N}}$ using $\bar{x}_{\mathcal{B}_1}$. Finally, we find a strictly feasible point to $\mathcal{F}$. This process is summarized in Algorithm 3.1. Furthermore, Algorithm 3.1 provides a constructive proof for Proposition 3.13 below.

Proposition 3.13. *Let $x \in \mathcal{F}$ be a solution such that $\mathrm{rank}\left(A(:, \mathrm{supp}(x))\right) = m$. Then, $\mathcal{F}$ has a strictly feasible point.*

Algorithm 3.1 Compute a Slater Point

Require: Given: A, $\bar{x} \in \mathcal{F}$ partitioned as in (3.3).

1: Choose any $\gamma \in \mathbb{R}^{|\mathcal{N}|}_{++}$ such that $\sum_{j=1}^{|\mathcal{N}|} \gamma_j < 1$.

2: Compute

$$\Theta = \begin{bmatrix} \bar{x}_{\mathcal{B}_1} & \cdots & \bar{x}_{\mathcal{B}_1} \end{bmatrix} \mathrm{Diag}(\gamma) - A_{\mathcal{B}_1}^{-1} A_{\mathcal{N}}.$$

3: Compute $\theta^* \in \mathbb{R}^{|\mathcal{N}|}_{++}$, where for each $j \in \{1, \ldots, |\mathcal{N}|\}$,

$$\theta_j^* := \begin{cases} \max(\Theta(:, j)) & \text{if } \max(\Theta(:, j)) \leq 1, \\ 1 & \text{otherwise.} \end{cases}$$

4: Set $x^\circ = \begin{pmatrix} \bar{x}_{\mathcal{B}_1} - (A_{\mathcal{B}_1})^{-1} A_{\mathcal{N}} \theta^* \\ \bar{x}_{\mathcal{B}_2} \\ \theta^* \end{pmatrix}$.

3.2.2 Towards an Exposing Vector; Phase I Part B and Strict Feasibility Testing

We now present an efficient preprocessing procedure for detecting identically 0 variables, obtaining exposing vectors and the facially reduced **LP**, i.e., given a **BFS** $\bar{x}$, we solve special subproblems using the simplex method with the initial point $\bar{x}$. By the end of the process, we obtain either

1. a certificate y that produces an exposing vector $A^T y$ (Slater condition fails), or
2. a strictly feasible point (Slater condition holds).

The process has two applications. First, since the only requirement of this process is the **BFS**, the procedure can be used as an extension of the phase-I of the two-phase simplex method to obtain the equivalent facially reduced problem that satisfies strict feasibility. Second, the procedure can be used as a post-optimum diagnosis. By recording a **BFS** with the smallest degree of degeneracy, we can improve tests for stability.

We now describe the proposed preprocessing method. Let $\mathcal{B}$ be a degenerate initial basis of $\mathcal{F}$ and let $\bar{x}$ be the **BFS** associated with $\mathcal{B}$. Without loss of generality, we assume that basic variables are located at the first m entries of $\bar{x}$. Let d be the degree of degeneracy of $\bar{x}$. We further assume that the degenerate basic variables are located at the first d entries of $\bar{x}$. We let $\mathcal{B}_0 := \{1, \dots, d\}$. We consider the following problem:

$$p_1^* = \max\{x_1 \,:\, Ax = b,\, x \geq 0\}. \tag{3.4}$$

We solve (3.4) using the simplex method from the initial **BFS** $\bar{x}$. That is, we do not need to perform the typical phase-I of the two-phase simplex method in order to find a feasible **BFS**. The optimal value p_1^* of (3.4) is clearly lower bounded by 0. We consider two cases below:

1. Suppose that $p_1^* > 0$. Then, the the variable x_1 is not an identically 0 variable, i.e., $1 \notin \mathcal{I}_0$.
2. Suppose that $p_1^* = 0$. Then, the variable x_1 is an identically 0 variable, i.e., $1 \in \mathcal{I}_0$. Let $\mathcal{B}^*$ be an optimal basis for (3.4). Then we have

   $$y^* = A(:, \mathcal{B}^*)^T e_1, \ \langle b, y^* \rangle = 0 \ \text{ and } A^T y^* \geq e_1. \tag{3.5}$$

   We note that the dual optimal solution y^* in (3.5) produces a solution to the auxiliary system (2.3). Therefore, we obtain a *nontrivial* exposing vector since $A^T y^*$ is not the zero vector. Clearly, the first variable x_1 is exposed by $A^T y^*$ since the first element of $A^T y^*$ is *positive*. Furthermore, if $|\operatorname{supp}(A^T y^*)| > 1$, then we find additional variables that are identically 0 in the feasible set. We can now delete the identified identically zero variables along with the corresponding columns of A. We then find and delete redundant rows to obtain a smaller **LP**.

Let $\{y^j\}$ be a collection of the certificates that are obtained from solving (3.4) with the index 1 replaced by $i \in \mathcal{B}_0$. Then $y^\circ = \sum_j y^j$ is also a certificate, i.e.,

$$A^T y^\circ = \sum_j A^T y^j \geq 0, \ A^T y^\circ \neq 0, \ \text{ and } \ \langle b, y^\circ \rangle = \sum_j \langle b, y^j \rangle = 0,$$

and we obtain a nontrivial exposing vector $A^T y^\circ$ for the system $\mathcal{F}$. By summarizing the two cases above, we obtain an efficient preprocessing method Algorithm 3.2.

The following allows for simplifications in Algorithm 3.2.

Lemma 3.14. *Let $\mathcal{B}$ be a basis containing an index i. The index i always remains in the basis $\mathcal{B}$ for the problem* (3.4).

HAESOL ♡ There is an error in this proof: should only include

Algorithm 3.2 Preprocessing Phase I Part B; Towards Strict Feasibility

Require: A **BFS** $\bar{x}$ with corresponding basis $\mathcal{B}$; set $\mathcal{B}_0 = \{i \in \mathcal{B} : \bar{x}_i = 0\}$

Initialize: $x^\circ = \bar{x}$, $y^\circ = 0 \in \mathbb{R}^m$, $\mathcal{J}_0 = \emptyset$, $\mathcal{B}_* \leftarrow \mathcal{B}_0$

while $\mathcal{B}_0 \neq \emptyset$ and $\mathcal{B}_* \neq \emptyset$ **do**

  Pick $i \in \mathcal{B}_0$; starting from the initial **BFS** $\bar{x}$, solve for primal-dual optima x^*, y^*

$$x^* = \text{argmax}_x\{x_i : Ax = b, x \geq 0\}, \quad p^* = x_i^* = b^T y^*$$

  $\mathcal{S} = \text{supp}(x^*)$

  $\mathcal{B}_* = \{i \in \mathcal{B}_* : x_i^* = 0\}$

  if $\mathcal{B}_0 \neq \emptyset$ and $\mathcal{B}_* \neq \emptyset$ **then**

    if $p^* = 0$ (strict feasibility fails) **then**

      Use dual certificate y^* to satisfy (2.3)

      $y^\circ \leftarrow y^\circ + y^*$

      $\mathcal{J}_0 \leftarrow \mathcal{J}_0 \cup (\text{supp}(A^T y^*) \cap \mathcal{B})$

      $\mathcal{B}_0 \leftarrow \mathcal{B}_0 \setminus \{\mathcal{S} \cup \mathcal{J}_0\}$

    else

      $\mathcal{B}_0 \leftarrow \mathcal{B}_0 \setminus \mathcal{S}$

    end if

    $x^\circ = \frac{1}{2}(x^\circ + x^*)$

  end if

end while

if $\mathcal{J}_0 \neq \emptyset$ **then**

  $z^\circ = A^T y^\circ$ (exposing vector)

  $\mathcal{R} \leftarrow$ redundant row indices of $A\,(:, \text{supp}(z^\circ)^c)$

  $A \leftarrow A(\mathcal{R}^c, \text{supp}(z^\circ)^c)$, $\; b \leftarrow b(\mathcal{R}^c)$

else

  Run Algorithm 3.1 with x° and $\det(A_{\mathcal{B}}) \neq 0$ (use x^* and $\mathcal{B}_*$, if $\mathcal{B}_* = \emptyset$)

end if

Without loss of generality, we let $i = 1$. We argue that 1 is not chosen to leave the basis. Let $y^* = (A_{\mathcal{B}}^T)^{-1}c_{\mathcal{B}}$. Suppose that the reduced cost at the index j is positive. Then

$$0 < \bar{c}_j = c_j - A_j^T y^* = -A_j^T y^* = -A_j^T (A_{\mathcal{B}}^T)^{-1} e_1 = -\bar{A}_{1j}.$$

Since $\bar{A}_{1j} < 0$, the index 1 is not chosen to leave the basis $\mathcal{B}$.

Proof. Without loss of generality, we let $i = 1$. We argue that 1 is not chosen to leave the basis. With $y^* = (A_{\mathcal{B}}^T)^{-1}c_{\mathcal{B}}$, it is sufficient to show that $A_{\mathcal{N}}^T y^* \geq 0$, i.e., that dual feasibility holds. To obtain a contradiction, suppose that $A_j^T < 0, j \in \mathcal{N}$. We observe the reduced cost at the index j:

$$\begin{array}{rcl} 0 < \bar{c}_j & = & c_j - A_j^T y^* \\ & = & -A_j^T y^* \\ & = & -A_j^T (A_{\mathcal{B}}^T)^{-1} e_1 \\ & = & -\bar{A}_{1j}. \end{array}$$

Since $\bar{A}_{1j} < 0$, the index 1 is not chosen to leave the basis $\mathcal{B}$. □

The following special case is of interest. Namely, no simplex pivoting steps are required to determine strict feasibility.

Theorem 3.15. *(preprocessing for degree of degeneracy 1) Let $\bar{x}$ be a **BFS** with the degree of degeneracy exactly one. Let $y^* = (A_{\mathcal{B}}^T)^{-1}c_{\mathcal{B}}$. Then strict feasibility fails if, and only if, y^* satisfies $A_{\mathcal{N}}^T y^* \geq 0$.*

Proof. Suppose that $\bar{x}$ is a degenerate **BFS** with basis $\mathcal{B}$. Without loss of generality, we assume $1 \in \mathcal{B}$ and 1 is the degenerate index. We consider the problem

$$p_1^* = \max\{x_1 \,:\, Ax = b,\, x \geq 0\}.$$

We note that $\langle b, y^* \rangle = 0$ since $\langle b, y^* \rangle$ is identical to the current objective value '0'. The backward direction is clear by Proposition 2.2. Now suppose that strict feasibility fails. Suppose to the contrary that $A_{\mathcal{N}}^T y^* \geq 0$ fails. Then there exists j such that $A_j^T y^* < 0,\ j \in \mathcal{N}$. Note that, by Lemma 3.14, that 1 is not chosen to leave the basis. Thus, there is an index $k \neq 1, k \in \mathcal{B}$ that leaves the basis. Since all other basic variables are positive, we obtain a positive step length and we improve the objective value, which yields a contradiction to $p_1^* = 0$. □

Upon the termination of Algorithm 3.2, we can always determine whether the system $\mathcal{F}$ has a strictly feasible point or not. Algorithm 3.2 terminates in a finite number of iterations since we remove at least one element from the set $\mathcal{B}_0$ in each iteration. We emphasize that we do not need to solve the auxiliary **LP**s for all $i \in \{1, \ldots, n\}$. We solve (3.4) only for the degenerate basic indices of the predetermined basis $\mathcal{B}$. However, upon the termination of Algorithm 3.2, we may not obtain face($\mathcal{F}$), the minimal face containing $\mathcal{F}$. Although the complete **FR** for **LP** can be performed in one iteration, one step termination is possible only when we find a solution y of (2.3) that is in the relative interior of the conjugate face of face($\mathcal{F}$). In this case, we can rerun Algorithm 3.2 with the facially reduced system. For finding an initial basis for the second trial, we may use the efficient basis recovery scheme [49, Chapter 7].

One of the nice features of Algorithm 3.2 is that we do not need to search for a new initial basis $\mathcal{B}$ for each iteration; the initial basis remains the same. Therefore, our approach can be directly employed after the standard phase-I of the two phase simplex method.

We do not need a lot of pivoting steps to determine if p_i^* is zero or positive. If $p_i^* = 0$, the initial $\mathcal{B}$ is indeed a basis that gives the optimal value. However the dual feasibility may not be obtained immediately[7]. Thus, there may be additional pivots required to obtain the dual feasibility. However, since the optimal value is obtained at $\mathcal{B}$, we do not expect that the optimal basis search to be time-consuming. For the case $p_i^* > 0$, the optimal value p_i^* does not need to be found. Hence once a basis that gives a positive support on i is found, we can terminate the maximization problem in Algorithm 3.2 immediately. We recall from Lemma 3.14 that the index i in (3.4) never leaves the basis.

We often get an exposing vector that reveals more than one element in the set $\mathcal{I}_0$ by solving (3.4). Without loss of generality, let $p_1^* = 0$ in (3.4) and let y^* be a dual feasible solution. Suppose $A^T y^* = e_1$, i.e., $A^T y^*$ only reveals exactly one exposed variable. Then $y^* \in \text{null}(A(:,2:n)^T)$. Since the data matrix A has more columns than the rows, $y^* \in \text{null}(A(:,2:n)^T)$ often implies that $y^* = 0$. If $y^* = 0$, we cannot obtain $A^T y^* = e_1$.

When an instance is large and have a **BFS** with a very large degree of degeneracy, one may adopt parallel computing for Algorithm 3.2 in order to reduce the total computation time. We note again that the initial basis remains the same throughout the iterations. Hence, solving (3.4) for individual $i \in \mathcal{B}_0$ can be performed independently. In fact, parallel computing can be used to obtain a strictly feasibility solution in Algorithm 3.1 as well; the weight vector γ can be chosen independently for each $j \in \mathcal{N}$.

3.3 Discussions

In this section we discuss the main result in Sections 3.1 and 3.2 and make connections to new results and known results in the literature.

3.3.1 Distance to Infeasibility

The *distance to infeasibility* is a measure of the smallest perturbations of the data (A,b) of a problem that renders the problem infeasible. In our setting, we can use the following simplification of the distance to infeasibility from [41] by restricting the perturbation to b, i.e., we can force infeasibility using only perturbation in b;

$$\text{dist}(b,\mathcal{F}=\emptyset) := \inf \left\{ \|b-\tilde{b}\| \ : \ \{x \in \mathbb{R}^n : Ax = \tilde{b}, \ x \geq 0\} = \emptyset \right\}.$$

Many interesting bounds, condition numbers, are shown in [41] under the assumption that the distance to infeasibility is positive and known. It is known that a positive distance to infeasibility of $\mathcal{F}$ implies that strict feasibility holds for $\mathcal{F}$; see e.g., [24,25]. The contrapositive of this statement is that, if strict feasibility fails for $\mathcal{F}$, then the distance to infeasibility is 0. We revisit this statement with the facially reduced system (2.5). We provide an elementary proof that there is an arbitrarily small perturbation for the data vector b of $\mathcal{F}$ that yields the set $\mathcal{F}$ infeasible, i.e., $\text{dist}(b,\mathcal{F}=\emptyset)=0$. Furthermore, we provide explicit perturbations that render the set $\mathcal{F}$ empty.

Suppose that $\mathcal{F}$ fails strict feasibility. Recall the representation (2.5) for $\mathcal{F}$. Let $AV = QR$ be a QR decomposition of AV, where $Q \in \mathbb{R}^{m\times m}$ orthogonal, $R \in \mathbb{R}^{m\times(n-s_z)}$ upper triangular. We write $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ so that $\text{range}(Q_1) = \text{range}(AV)$. Then, by the orthogonality of Q, we have

$$Ax = AVv = b \iff Q^T Ax = Rv = Q^T b.$$

[7]If we have a nondegenerate initial basis, then the dual feasibility is immediately obtained. However, our initial basis is degenerate.

Since AV is a rank deficient matrix (see Lemma 2.3), the upper triangular matrix R is of the form

$$R = \begin{bmatrix} \bar{R} \\ 0 \end{bmatrix} \in \mathbb{R}^{m\times(n-s_z)} \text{ and } \bar{R} \in \mathbb{R}^{\operatorname{rank}(AV)\times(n-s_z)} \text{ with nonzero diagonal.} \tag{3.6}$$

Since $b \in \operatorname{range}(AV)$, the last $m - \operatorname{rank}(AV)$ entries of $Q^T b$ are equal to 0, i.e.,

$$Q^T b = \begin{pmatrix} Q_1^T b \\ Q_2^T b \end{pmatrix} = \begin{pmatrix} Q_1^T b \\ 0 \end{pmatrix}.$$

Consequently, the unrealized implicit non-surjuectivity produces the system

$$\begin{bmatrix} \bar{R} \\ 0 \end{bmatrix} v = \begin{pmatrix} Q_1^T b \\ 0 \end{pmatrix}, \ v \in \mathbb{R}^{n-s_z}_+. \tag{3.7}$$

Any perturbation on the last $m-\operatorname{rank}(AV)$ equations in (3.7) that causes the system inconsistency renders the system (3.7) infeasible while maintaining the dimension of $\operatorname{relint}(\mathcal{F})$. For instance, replacing the right-hand-side vector in (3.7) by $\begin{pmatrix} Q_1^T b \\ \xi \end{pmatrix}$ with any nonzero vector $\xi \in \mathbb{R}^{m-\operatorname{rank}(AV)}$ renders (3.7) infeasible. Replacing the data matrix in (3.7) by $\begin{bmatrix} \bar{R} \\ \Phi \end{bmatrix}$ for which Φ contains a positive row vector also renders (3.7) infeasible.

We now present a class of perturbations of b that maintains the feasibility of the set $\mathcal{F}$ as well as a special perturbation of b that forces $\mathcal{F}$ to be infeasible. Such perturbations can be found using linear combinations of the columns of Q_1 or Q_2, respectively. We relate this observation to the solution of the auxiliary system (2.3) in the proof of Proposition 3.16 below.

Proposition 3.16. *Suppose that strict feasibility fails for $\mathcal{F}$, and let $\mathcal{F}$ have the representation (2.5). Then the following hold.*

1. *For all $\Delta b \in \operatorname{range}(AV)$ with sufficiently small norm, the set $\{x \in \mathbb{R}^n_+ : Ax = b + \Delta b\}$ is feasible.*
2. *Let $\bar{y} \in \mathbb{R}^m$ be a solution to the auxiliary system (2.3). Then perturbing the right-hand-side vector b of $\mathcal{F}$ in the direction $\bar{y}$ makes the system $\mathcal{F}$ infeasible.*

Proof. Let Δb be any perturbation in $\operatorname{range}(AV)$. Let $QR = AV$ be a QR decomposition of AV. In particular, let R have the form (3.6) and $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ so that $\operatorname{range}(Q_1) = \operatorname{range}(AV)$. Let ϵ be a sufficiently small scalar. Then

$$Ax = AVv = b + \epsilon\Delta b \iff Rv = Q^T b + \epsilon Q^T \Delta b \iff \bar{R}v = Q_1^T b + \epsilon Q_1^T \Delta b. \tag{3.8}$$

The last equivalence holds since $Ax = b$ and $\Delta b \in \operatorname{range}(AV) = \operatorname{range}(Q_1)$. Since the system $\bar{R}v = Q_1^T b$ satisfies strict feasibility, the distance to infeasibility of this system is positive. Thus, the perturbed system $\bar{R}v = Q_1^T b + \epsilon Q_1^T \Delta b$ remains feasible. Therefore, by (3.8), perturbing $\mathcal{F}$ along the direction $\Delta b \in \operatorname{range}(AV)$ maintains the feasibility and this concludes the proof for Item 1.

For Item 2 we show that perturbing b with $\Delta b = \bar{y}$ renders $\mathcal{F}$ infeasible, where $\bar{y}$ is a solution to the system (2.3). By Proposition 2.2 and (2.6), the nonzero vector $\bar{y} \in \mathbb{R}^m$ is in $\operatorname{null}((AV)^T)$. Then we have

$$\bar{y} \in \operatorname{range}(AV)^\perp = \operatorname{range}(Q_2) \implies \bar{y} = Q_2\bar{u} \text{ for some nonzero } \bar{u}.$$

We recall Farkas' lemma:

$$\{y \in \mathbb{R}^m : A^T y \geq 0,\ \langle b, y\rangle < 0\} \neq \emptyset \implies \mathcal{F} = \emptyset.$$

Now, for any $\epsilon > 0$, setting $\Delta b_\epsilon = -\epsilon \bar{y}$ yields

$$A^T \bar{y} \geq 0,\ \langle b, \bar{y}\rangle = 0 \implies A^T \bar{y} \geq 0,\ \langle b + \Delta b_\epsilon, \bar{y}\rangle < 0. \tag{3.9}$$

Hence, by letting $\epsilon \to 0^+$, we see that the distance to infeasibility, $\operatorname{dist}(b, \mathcal{F} = \emptyset)$, is equal to 0. □

We emphasize that the result

$$\mathcal{F} \text{ fails strictly feasibility } \implies \operatorname{dist}((A, b), \mathcal{F} = \emptyset) = 0$$

gives rise to the second step (2.7) of **FR** discussed in Section 2.2. We note that the instability discussed in this section essentially originates from the observation made in Lemma 2.3, i.e., redundant equalities arise in the facially reduced system. Facially reduced system allows us to exploit the root of potential instability when the problem data A or b is perturbed. Although the distance to infeasibility is 0 in the absence of strict feasibility, Proposition 3.16 suggests that a carefully chosen perturbation of b does not have an impact on the feasibility of $\mathcal{F}$. We provide a related numerical experiment in Section 4.1.4 below.

The distance to infeasibility directly impacts the measure of well-posedness of the problem, [24, 25, 42]. Given the pair $d = (A, b)$ of the data for an instance $(\mathcal{P})$, the *condition measure* of $(\mathcal{P})$ is defined by

$$\mathcal{C}(d) := \frac{\|d\|}{\inf\{\|\Delta d\| : d + \Delta d \text{ yields } (\mathcal{P}) \text{ infeasible}\}}.$$

The value $\mathcal{C}(d)$ is a measure of well-posedness of the problem $(\mathcal{P})$. Since $\operatorname{dist}(b, \mathcal{F} = \emptyset) = 0$, we have $\mathcal{C}(d) = \infty$. Namely, when strict feasibility fails for $(\mathcal{P})$, the problem is ill-posed.

3.3.2 Applications to Known Characterizations for Strict Feasibility

There are some known characterizations for strict feasibility of $\mathcal{F}$. Using these characterizations we can obtain extensions of Theorem 3.1, Theorem 3.5, and Corollary 3.8.

The dual $(\mathcal{D})$ of $(\mathcal{P})$ is

$$(\mathcal{D}) \qquad \max_{y,s} \left\{ b^T y \ :\ A^T y + s = c,\ s \geq 0 \right\}. \tag{3.10}$$

It is known that strict feasibility fails for $\mathcal{F}$ if, and only if, the set of optimal solutions for the dual $(\mathcal{D})$ is unbounded; see e.g., [49, Theorem 2.3] and [27]. Then Corollary 3.17 follows.

Corollary 3.17. 1. *Suppose that the set of optimal solutions for the dual $(\mathcal{D})$ is unbounded. Then every basic feasible solution to $\mathcal{F}$ is degenerate.*

2. *Suppose that there exists a nondegenerate basic feasible solution to $\mathcal{F}$. Then the set of optimal solutions for the dual $(\mathcal{D})$ is bounded.*

It is known that strict feasibility holds for $\mathcal{F}$ if, and only if, $b \in \operatorname{relint}(A(\mathbb{R}^n_+))$, where relint denote the relative interior; see e.g., [22, Proposition 4.4.1]. Then if one finds a set of indices $\mathcal{I} \subset \{1, \ldots, n\}$ such that $A(:, \mathcal{I})$ is nonsingular and $A(:, \mathcal{I})z = b$ has a solution z with positive entries, then $b \in \operatorname{relint}(A(\mathbb{R}^n_+))$.

3.3.3 Applications to Obtain a Strictly Complementary Primal-Dual Solution

In this section we present an application of Algorithm 3.1 for obtaining a strictly complementary primal-dual optimal solution.

Let (x^*, y^*, s^*) be an optimal triple for the standard primal-dual **LP** pair. Let $\mathcal{B}^* \cup \mathcal{N}^* = \{1, \ldots, n\}$ be the strict complementary partition of the primal-dual optimal pair. The existence of such a partition is guaranteed by the Goldman-Tucker theorem [28] and the partition $\mathcal{B}^* \cup \mathcal{N}^*$ is unique. For the first application of Algorithm 3.1, we provide a method for obtaining a strict complementary primal-dual solution when the primal optimal solution x^* is nondegenerate or the submatrix $A(:, \operatorname{supp}(x^*))$ of A has rank m. To elaborate, we list the two cases where Algorithm 3.1 can be used to obtain maximal complementary solutions.

1. Let x^* be a nondegenerate (optimal) basic feasible solution. Then, $\operatorname{supp}(s^*) = \mathcal{N}^*$ and $\operatorname{supp}(x^*)$ can be extended to complete $\mathcal{B}^*$;
2. Let x^* be an optimal solution such that $A(:, \operatorname{supp}(x^*))$ is full row rank. Then, $\operatorname{supp}(s^*) = \mathcal{N}^*$ and $\operatorname{supp}(x^*)$ can be extended to complete $\mathcal{B}^*$.

Suppose that we are given a primal-dual optimal solution (x^*, y^*, s^*) of the form

$$\begin{bmatrix} A_{\mathcal{B}} & A_{\mathcal{J}} & A_{\mathcal{N}} \end{bmatrix} \begin{pmatrix} x_{\mathcal{B}} \\ x_{\mathcal{J}} \\ x_{\mathcal{N}} \end{pmatrix} = b, \text{ where } \operatorname{rank}(A_{\mathcal{B}}) = m, \begin{pmatrix} x_{\mathcal{B}} \\ x_{\mathcal{J}} \\ x_{\mathcal{N}} \end{pmatrix} \begin{matrix} > \\ = \\ = \end{matrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} s_{\mathcal{B}} \\ s_{\mathcal{J}} \\ s_{\mathcal{N}} \end{pmatrix} \begin{matrix} = \\ = \\ > \end{matrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{3.11}$$

We claim that $\mathcal{N}^* = \operatorname{supp}(s^*)$. That is, the support of the current dual optimal solution s^* is maximal and hence we obtain the strict complementary partition for free. We rewrite the system $Ax = b$ of (3.11) as

$$\begin{bmatrix} A_{\mathcal{B}_1} & A_{\mathcal{B}_2} & A_{\mathcal{J}} \end{bmatrix} \begin{pmatrix} x_{\mathcal{B}_1} \\ x_{\mathcal{B}_2} \\ x_{\mathcal{J}} \end{pmatrix} = b, \quad \text{where } A_{\mathcal{B}} = \begin{bmatrix} A_{\mathcal{B}_1} & A_{\mathcal{B}_2} \end{bmatrix}, \ x_{\mathcal{B}} = \begin{pmatrix} x_{\mathcal{B}_1} \\ x_{\mathcal{B}_2} \end{pmatrix} \text{ and } \operatorname{rank}(A_{\mathcal{B}_1}) = m.$$

Then, by replacing $\mathcal{N}$ in Algorithm 3.1 by $\mathcal{J}$, we can endow positive weights to $x_{\mathcal{J}}$ while maintaining the primal feasibility. Since we maintain the feasibility of the primal-dual solution without violating the complementarity, we maintain the optimality.

3.3.4 Lack of Strict Feasibility in the Dual

Recall Remark 2.1 that redundant constraints can result in poor behaviour for interior point methods. Moreover, complementary slackness means we get dual variables fixed at 0. This is one motivation for considering **FR** on the dual $(\mathcal{D})$; see (3.10).

We denote the feasible set of the dual $(\mathcal{D})$ by

$$\mathcal{G} := \{(y, s) \in \mathbb{R}^m \oplus \mathbb{R}^n_+ \ : \ A^T y + s = c\}. \tag{3.12}$$

The facial reduction arguments applied to the dual are parallel to the ones given in Section 2.2. Hence, we provide a short derivation for the facially reduced system for $\mathcal{G}$. We also conclude that the absence of strict feasibility for $\mathcal{G}$ implies dual degeneracy at all basic feasible solutions.

The following lemma is the theorem of the alternative applied to the set $\mathcal{G}$.

Lemma 3.18. *[13, Theorem 3.3.10](theorem of the alternative in dual form) Let $\mathcal{G}$ in (3.12) be feasible. Then, exactly one of the following statements holds:*

1. *There exists $(y,s) \in \mathbb{R}^m \oplus \mathbb{R}^n_{++}$ with $A^T y + s = c$, i.e., strict feasibility holds for $\mathcal{G}$;*
2. *There exists $w \in \mathbb{R}^n$ such that*

$$0 \neq w \in \mathbb{R}^n_+, \ Aw = 0 \ \text{ and } \ \langle c, w\rangle = 0. \tag{3.13}$$

We recall that the vector $A^T y$ in (2.4) is an exposing vector to the set $\mathcal{F}$. Let w be a solution to the auxiliary system (3.13). Similarly, the vector w plays the role of an exposing vector for $\mathcal{G}$:

$$\forall (y,s) \in \mathcal{G}, \ \text{ it holds } \ \langle w, s\rangle = \langle w, c - A^T y\rangle = \langle c, w\rangle - \langle Aw, y\rangle = 0 - \langle 0, y\rangle = 0.$$

We let

$$\mathcal{I}_w = \{1,\ldots,n\} \setminus \operatorname{supp}(w), \ U = I(:,\mathcal{I}_w) \ \text{ and } \ s_w = |\operatorname{supp}(w)|.$$

Then, the facially reduced system of $\mathcal{G}$ is given by

$$\mathcal{G} = \left\{ (y,s) \in \mathbb{R}^m \oplus \mathbb{R}^n_+ \ : \ \begin{bmatrix} A^T & I \end{bmatrix} \begin{pmatrix} y \\ s \end{pmatrix} = c \right\} = \left\{ (y,u) \in \mathbb{R}^m \oplus \mathbb{R}^{n-s_w}_+ \ : \ \begin{bmatrix} A^T & U \end{bmatrix} \begin{pmatrix} y \\ u \end{pmatrix} = c \right\}. \tag{3.14}$$

The notion of degeneracy in Section 2.1 naturally extends to an arbitrary polyhedron, e.g., see [5, Section 2]. For a general polyhedron $P \subseteq \mathbb{R}^n$, a point p in P is called a *basic solution* if there are n linearly independent active constraints at p. In addition, if there are more than n active constraints at the point $p \in P$, then the point p is called *degenerate*. Using this definition of the degeneracy, we now show that the absence of strict feasibility for $\mathcal{G}$ implies that every basic solution of $\mathcal{G}$ is degenerate.

We show that the facially reduced system in (3.14) contains a redundant constraint. Let w be a solution to the system (3.13), i.e., w is an exposing vector for $\mathcal{G}$. Then we have

$$\begin{bmatrix} A \\ U^T \end{bmatrix} w = \begin{bmatrix} Aw \\ U^T w \end{bmatrix} = \begin{bmatrix} 0_m \\ 0_{n-s_w} \end{bmatrix}.$$

In other words, there is a nontrivial row combination of $\begin{bmatrix} A^T & U \end{bmatrix}$ that yields the 0 vector, i.e., there exists a redundant row in $\begin{bmatrix} A^T & U \end{bmatrix}$. Hence, the facially reduced system contains a redundant constraint. The redundancy immediately implies the dual degeneracy; for any basic solution of $\mathcal{G}$, there always exists a redundant equality in $\begin{bmatrix} A^T & I \end{bmatrix} \begin{pmatrix} y \\ s \end{pmatrix} = c$.

3.3.5 Lack of Strict Feasibility and Interior Point Methods

Many interior point algorithms are derived from the optimality conditions (KKT conditions) using the primal ($\mathcal{P}$) and the dual ($\mathcal{D}$). And many practical interior point methods find the search direction d by solving the so-called normal equation, a square system $ADA^T d = r$, where D is a diagonal matrix with positive diagonal and r is some residual; see e.g., [49, Chapter 11]. The diagonal of D consists of some element-wise product of the primal variable x and the dual slack variable s. We have shown that the lack of strict feasibility for $\mathcal{F}$ makes all vertices of $\mathcal{F}$ degenerate. Thus, all vertices that form the optimal face of ($\mathcal{P}$) are also degenerate. Hence, primal or dual degeneracy can cause the diagonal of D to have a deficient number of nice positives near the optimum. Thus the normal equation may be ill-conditioned as it gets near the optimum, i.e., numerical stability could be hard to achieve. We present related numerics in Section 4.1.[8]

4 Numerics

We now provide empirical evidence that **FR** is indeed a useful preprocessing tool for reducing the size of problems as well as for improving the *conditioning*. We do this first for interior point methods and then for simplex methods. In particular, this provides empirical evidence that lack of strict feasibility is equivalent to implicit singularity. All the numerical tests are performed using MATLAB version 2021a on Dell XPS 8940 with 11th Gen Intel(R) Core(TM) i5-11400 @ 2.60GHz 2.60 GHz with 32 Gigabyte memory. We use three different solvers in our tests: (i) *linprog* from MATLAB[9]; (ii) *SDPT3*[10]; and (iii) *MOSEK*[11]. MATLAB version 2021a is used to access all the solvers for the tests, and we use their default settings for stopping criteria. Note that MOSEK has a preprocessing option.[12]

4.1 Empirics with Interior Point Methods

In this section we compare the behaviour for finding near-optimal points with instances that do and do not satisfy strict feasibility. More specifically, given a near optimal primal-dual point $(x^*, s^*) \in \mathbb{R}^n_{++} \oplus \mathbb{R}^n_{++}$ obtained from an interior point solver, we observe the condition number, i.e., the ratio of largest to smallest eigenvalues of the normal matrix at (x^*, s^*):

$$\kappa\left(AD^*A^T\right), \quad \text{where } D^* = \operatorname{Diag}(x^*)\operatorname{Diag}(s^*)^{-1}. \tag{4.1}$$

We show that instances that do not have strictly feasible points tend to have significantly larger condition numbers of the normal equation near the optimum. We also present a numerical experiment on perturbations of the right-hand-side vector b.

4.1.1 Generating LPs without Strict Feasibility

Given $m, n, r \in \mathbb{N}$, we construct the data $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ to satisfy (2.3) with r as the dimension of the relative interior of $\mathcal{F}$, $\operatorname{relint}(\mathcal{F})$.

1. Pick any $0 \neq y \in \mathbb{R}^m$. Let

$$\{y\}^\perp = \operatorname{span}\{a_i\}_{i=1}^{m-1} \quad (= \operatorname{null}(y^T)).$$

   We let $R \in \mathbb{R}^{(m-1)\times r}$ be a random matrix, and get

$$A_1 := \begin{bmatrix} a_1 & \dots & a_{m-1} \end{bmatrix} R \in \mathbb{R}^{m\times r}, \quad A_1^T y = 0 \in \mathbb{R}^r.$$

2. Pick any $\hat{v} \in \mathbb{R}^r_{++}$ and set $b = A_1\hat{v}$. We note that $y^T A_1 = 0$ and $\langle b, y\rangle = 0$.

[8]The survey [32] addresses the effect of degeneracy on the performance of interior point methods.
[9]https://www.mathworks.com/. Version 9.10.0.1669831 (R2021a) Update 2.
[10]https://www.math.cmu.edu/~reha/sdpt3.html, version SDPT3 4.0.
[11]https://www.mosek.com/. Version 8.0.0.60.
[12]MOSEK has a presolve with five steps that includes eliminating fixed variables. However, it is clear from the empirical evidence that the variables fixed at 0 are not found.

3. Pick any matrix $A_2 \in \mathbb{R}^{m\times(n-r)}$ satisfying $(y^TA_2)_i \neq 0,\ \forall i$. If there exists i such that $(y^TA_2)_i < 0$, then change the sign of the i-th column of A_2 so that we conclude

$$(A_2^Ty) \in \mathbb{R}^{n-r}_{++}.$$

4. We define the matrix $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \in \mathbb{R}^{m\times n}$. Then $\{x \in \mathbb{R}^n_+ : Ax = b\}$ is a polyhedron with a feasible point $\hat{x} = [\hat{v}; 0]$ having r number of positives. The vector y is a solution for the system (2.3):

$$0 \lneqq z = A^Ty = \begin{pmatrix} A_1^Ty = 0 \\ A_2^Ty > 0 \end{pmatrix},\ b^Ty = 0.$$

   We then randomly permute the columns of A to avoid the zeros always being at the bottom of the feasible variables x.

For the empirics, we construct the objective function c^Tx of $(\mathcal{P})$ as follows. We choose any $\bar{s} \in \mathbb{R}^n_{++}, \bar{y} \in \mathbb{R}^m$ and set $c = A^T\bar{y} + \bar{s}$. Then we have the data for the primal-dual pair of **LP**s and the primal *fails* strict feasibility:

$$(\mathcal{P}_{(A,b,c)}) \quad \min\{\ c^Tx : Ax = b,\ x \geq 0\ \} \quad \text{and} \quad (\mathcal{D}_{(A,b,c)}) \quad \max\{\ b^Ty : A^Ty + s = c,\ s \geq 0\ \}.$$

We note that by choosing $\bar{s} \in \mathbb{R}^n_{++}$, the dual problem $(\mathcal{D}_{(A,b,c)})$ has a strictly feasible point. In order to generate instances with strictly feasible points, we maintain the same data A, c used for the pair $(\mathcal{P}_{(A,b,c)})$ and $(\mathcal{D}_{(A,b,c)})$. We only redefine the right-hand-side vector by $\bar{b} = Ax^\circ$, where $x^\circ \in \mathbb{R}^n_{++}$:

$$(\bar{\mathcal{P}}_{(A,\bar{b},c)}) \quad \min\{\ c^Tx : Ax = \bar{b},\ x \geq 0\ \} \quad \text{and} \quad (\bar{\mathcal{D}}_{(A,\bar{b},c)}) \quad \max\{\ \bar{b}^Ty : A^Ty + s = c,\ s \geq 0\ \}.$$

The facially reduced instances of $(\mathcal{P}_{(A,b,c)})$ are denoted by $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$. They are obtained by discarding the variables that are identically 0 in the feasible set $\mathcal{F}$ and the redundant constraints. In other words, the affine constraints of $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$ are of the form (2.7).

4.1.2 Condition Numbers

In order to illustrate the differences in condition numbers of the normal matrices, we solve the three families of instances:
(i) $(\mathcal{P}_{(A,b,c)})$, strictly feasible fails; (ii) $(\bar{\mathcal{P}}_{(A,\bar{b},c)})$, strictly feasible holds; (iii) $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$, facially reduced instances of $(\mathcal{P}_{(A,b,c)})$.

In Figure 4.1 we use a *performance profile* [20,31] to observe the overall behaviour on different families of instances using the three solvers. The performance profile provides a useful graphical comparison for solver performances. Figure 4.1 displays the performance profile on the condition numbers of the normal matrix AD^*A^T near optimal points from different solvers. We generate 100 instances for each family that have $\dim(\text{relint}(\mathcal{F})) \in [300, 1350]$. The instance sizes are fixed with $(m, n) = (500, 1500)$. The vertical axis in Figure 4.1 represents the statistics of the performance ratio on $\kappa\left(AD^*A^T\right)$, the condition number of normal matrix near optimum (x^*, s^*); see (4.1). The solid lines in Figure 4.1 represent the performance of the instances $(\mathcal{P}_{(A,b,c)})$ that fail strict feasibility. They show that the condition numbers of the normal matrices near optima are significantly higher when strict feasibility fails. That is, when strict feasibility fails for $\mathcal{F}$, the matrix AD^*A^T is more ill-conditioned and it is difficult to obtain search directions of high accuracy. We also observe that facially reduced instances yield smaller condition numbers near optima. We note that the instances $(\mathcal{P}_{(A,b,c)})$ and $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$ are equivalent.

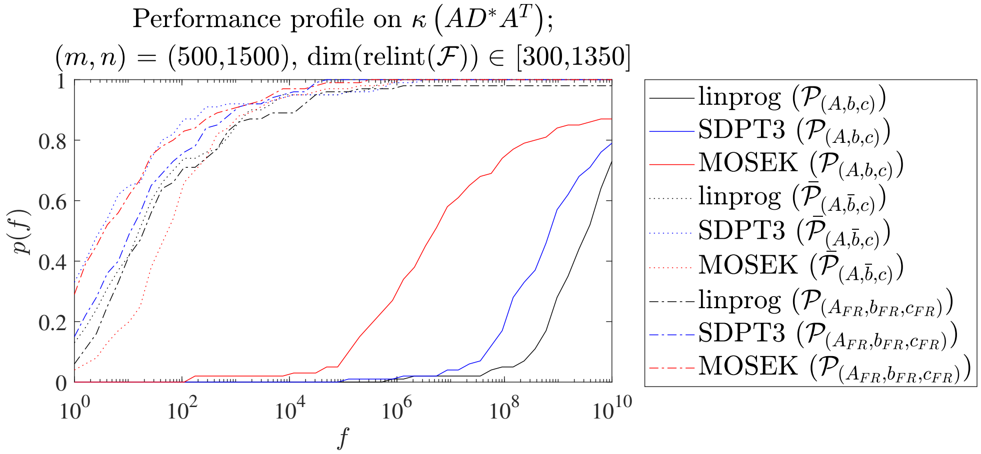

Figure 4.1: Performance profile on $\kappa\left(ADA^T\right)$ with(out) strict feasibility near optimum; various solvers

4.1.3 Stopping Criteria

We now use the three solvers to observe the accuracy of the first-order optimality conditions (KKT conditions) and the running times, for the instances $(\mathcal{P}_{(A,b,c)})$ and $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$, see Table 4.1. We test the average performance of 10 instances of the size $(n,m,r) = (3000, 500, 2000)$. The headers used in Table 4.1 provide the following. Given solver outputs (x^*, y^*, s^*), the header 'KKT' exhibits the average of the triple consisting of the primal feasibility, dual feasibility and complementarity;

$$\text{KKT} = \left(\frac{\|Ax^* - b\|}{1+\|b\|}, \frac{\|A^T y^* + s^* - c\|}{1+\|c\|}, \frac{\langle x^*, s^* \rangle}{n} \right).$$

The headers 'iter' and 'time' in Table 4.1 refer to the average of the number of iterations and the running time in seconds, respectively.

		Non-Facially Reduced System	Facially Reduced System
linprog	KKT	(2.44e-15, 2.05e-12, 3.18e-09)	(5.85e-16, 4.74e-16, 9.22e-09)
	iter	22.30	17.90
	time	2.34	0.81
SDPT3	KKT	(8.11e-10, 7.55e-12, 5.65e-02)	(1.43e-11, 3.67e-16, 4.38e-06)
	iter	25.50	19.30
	time	1.73	0.70
mosek	KKT	(7.52e-09, 1.80e-15, 3.27e-06)	(3.85e-09, 3.69e-16, 1.19e-06)
	iter	40.30	10.20
	time	1.40	0.35

Table 4.1: Average of KKT conditions, iterations and time of (non)-facially reduced problems

From Table 4.1 we observe that facially reduced instances provide significant improvement in first-order optimality conditions, the number of iterations and the running times for all solvers in general. We note that the instances $(\mathcal{P}_{(A,b,c)})$ and $(\mathcal{P}_{(A_{FR},b_{FR},c_{FR})})$ are equivalent. Hence, our empirics show that performing facial reduction as a preprocessing step not only improves the solver running time but also the *quality* of solutions.

4.1.4 Distance to Infeasibility

In this section we present empirics that illustrate the impact of perturbations of the right-hand-side b when strict feasibility fails. We recall, from Proposition 3.16, that there exists an arbitrarily small perturbation of the right-hand-side vector b of $\mathcal{F}$ that renders the set $\mathcal{F}$ infeasible, i.e., $\text{dist}(b, \mathcal{F} = \emptyset) = 0$. Moreover, the vector $\Delta b = y$ that satisfies the auxiliary system (2.3) is a perturbation that makes the set $\mathcal{F}$ empty; see (3.9).

We follow the steps in Section 4.1.1 to generate instances of the order $(n, m) = (1000, 200)$ and $r = \text{relint}(\mathcal{F}) = 900$. The objective function $c^T x$ is chosen as presented in Section 4.1.1. For the fixed (n, m, r), we generate 10 instances and observe the average performance of these instances as we gradually increase the magnitude of the perturbation. We recall the matrix AV from (2.5). We use two types of perturbations for b;

$$\Delta b, \text{ where } \Delta b \in \text{range}(AV)^{\perp}, \quad \Delta \bar{b}, \text{ where } \Delta \bar{b} \in \text{range}(AV).$$

We choose Δb to be the vector y that satisfies (2.3). For $\Delta \bar{b}$, we choose AVd, where $d \in \mathbb{R}^r$ is a randomly chosen vector. As we increase $\epsilon > 0$, we observe the performance of the two families of the systems

$$Ax = b_\epsilon := b - \epsilon \Delta b \text{ and } Ax = \bar{b}_\epsilon := b - \epsilon \Delta \bar{b}.$$

We use the interior point method from MATLAB's linprog for the test. Figure 4.2 contains the average of the first-order optimality conditions evaluated at the solver outputs (x^*, y^*, s^*) of these instances; primal feasibility, dual feasibility and the complementarity.

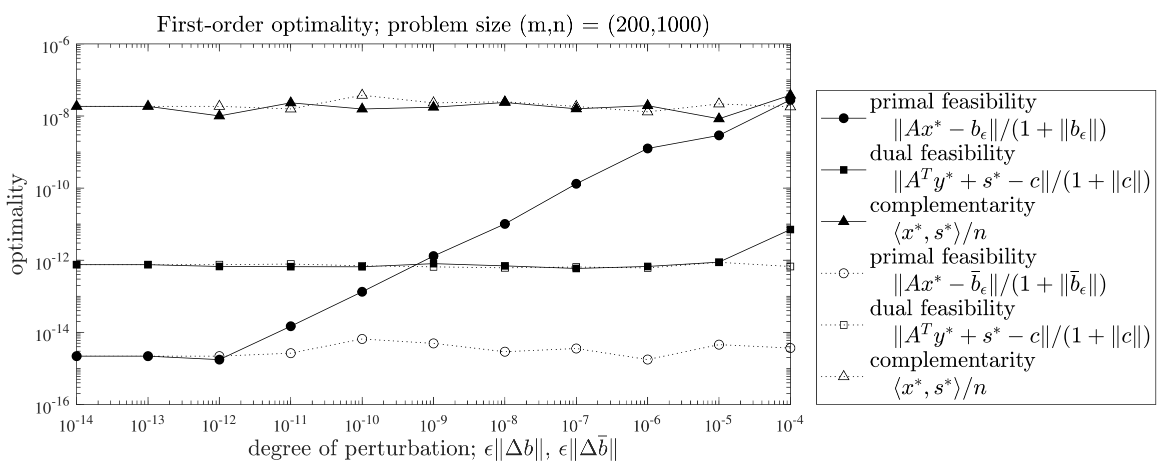

Figure 4.2: Changes in the first-order optimality condition as the perturbation of b increases

The horizontal axis of Figure 4.2 indicates the degree of the perturbation imposed on the right-hand-side vector b, $\epsilon\|\Delta b\|$ and $\epsilon\|\Delta \bar{b}\|$. The vertical axis indicates the individual component of the first-order optimality. From Figure 4.2, we observe that the KKT conditions with the perturbation $\Delta \bar{b}$ display a steady performance regardless of the perturbation degree; see the markers ○,□,△ with the dotted lines. In contrast, the markers ●,■,▲ in Figure 4.2 exhibit the performance of the instances that are perturbed with Δb and they display a different performance. In particular, we see that the relative primal feasibility $\|Ax^* - b_\epsilon\|/(1+\|b_\epsilon\|)$, marked with ●, consistently increases as the perturbation magnitude $\epsilon\|\Delta b\|$ increases when strict feasibility fails for $\mathcal{F}$.

4.1.5 Empirics on Singular Values and IPS

We partition the matrix $A = \begin{bmatrix} P_{\bar{m}}AV & A_{\mathcal{I}_0} \\ R_{AV} & R_{\mathcal{I}_0} \end{bmatrix}$, where $A_{\mathcal{I}_0}$ corresponds to the submatrix of A associated with the index set $\mathcal{I}_0$. The submatrix $\begin{bmatrix} R_{AV} & R_{\mathcal{I}_0} \end{bmatrix}$ refers to the rows of A that are implicitly redundant due the lack of strict feasibility. Let $D_c = \operatorname{Diag}(x_c)\operatorname{Diag}(s_c)^{-1}$, where x_c and s_c are the current primal-dual iterates. The ill-conditioning of the matrix AD_cA^T under the degeneracy is discussed in [32] in terms of the lack of nice positives of the diagonal D_c. The lack of strict feasibility also gives rise to this phenomenon in the sense that all the **BFS**s are degenerate and the relative interior of the optimal face may fail to produce nice positives on the diagonal D_c near the optima. In this section we argue that, under the lack of strict feasibility, the ill-conditioning of the matrix AD_cA^T not only originates form D_c but also originates from the rows of A. We relate this argument with IPS (see Definition 2.5) and present our numerical experiment. That is, a large IPS is a good indicator for ill-conditioning in the sense that it provides a lower bound to the number of implicit redundant constraints.

Let x^* be an optimal solution and let D^* be diagonal matrix defined in (4.1). As $x_c \to x^*$, i.e., as the iterates get closer to the feasible set $\mathcal{F}$ when an infeasible start interior point method is used, we observe the behaviour $AD_cA^T \to AD^*A^T$ below:

$$\begin{array}{rcl} AD_cA^T & = & \begin{bmatrix} P_{\bar{m}}AV & A_{\mathcal{I}_0} \\ R_{AV} & R_{\mathcal{I}_0} \end{bmatrix} \operatorname{Diag}(x_c)\operatorname{Diag}(s_c)^{-1} \begin{bmatrix} P_{\bar{m}}AV & A_{\mathcal{I}_0} \\ R_{AV} & R_{\mathcal{I}_0} \end{bmatrix}^T \\ \to \quad AD^*A^T & = & \begin{bmatrix} P_{\bar{m}}AV & A_{\mathcal{I}_0} \\ R_{AV} & R_{\mathcal{I}_0} \end{bmatrix} \begin{bmatrix} D^*_{AV} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_{\bar{m}}AV & A_{\mathcal{I}_0} \\ R_{AV} & R_{\mathcal{I}_0} \end{bmatrix}^T \\ & = & \begin{bmatrix} (P_{\bar{m}}AV)D^*_{AV}(P_{\bar{m}}AV)^T & (P_{\bar{m}}AV)D^*_{AV}R^T_{AV} \\ R_{AV}D^*_{AV}(P_{\bar{m}}AV)^T & R_{AV}D^*_{AV}R^T_{AV} \end{bmatrix}. \end{array}$$

We recall from Lemma 2.3 that the rows of R_{AV} are linear combination of the rows of $P_{\bar{m}}AV$. Therefore, AD^*A^T is ill-conditioned if strict feasibility fails. If R_{AV} contains many rows, AD^*A^T has many '0' singular values.

We generated instances with different settings for IPS $= 1, 5$ and 10. We recall the generation for the vector y and A_2 in Section 4.1.1. For generating and instance with IPS > 1, we generated $Y_c = \operatorname{blkdiag}(y^1, \ldots, y^{\text{IPS}}) \in \mathbb{R}^{m \times \text{IPS}}$ and $A_2 = \operatorname{blkdiag}(A_2^1, \ldots, A_2^{\text{IPS}})$ of appropriate dimension in order to produce the exposing vector $A_2^T \sum_{j=1}^{\text{IPS}} Y_c(:, j) \geq 0$. Each column of Y_c serves as a vector satisfying (2.3).

HAESOL ♡ perhaps better to use maxSD than IPS

Let $\sigma_{\max}(AD^*A^T)$ be the maximum singular value of AD^*A^T. We count the number of singular values of AD^*A^T that are smaller than $10^{-8} \cdot \sigma_{\max}(AD^*A^T)$. In Table 4.2 below, we report the cardinality of

$$\Sigma_0 := \{i : \sigma_i(AD^*A^T) < \sigma_{\max}(AD^*A^T)\}.$$

We test the average performance on the 20 instances of the fixed size $(n, m, r) = (3000, 500, 2000)$. We display the average number of $|\Sigma_0|$. We see from Table 4.2 a larger IPS value produces a greater number of small singular values. When there is a significant number of redundant constraints, it is more difficult to obtain a good search direction due to a large number of relatively small singular values.

		ips = 1	ips = 5	ips = 10
linprog	$\lvert\Sigma_0\rvert$	4.10	8.65	13.10
SDPT3	$\lvert\Sigma_0\rvert$	4.75	8.00	34.65
MOSEK	$\lvert\Sigma_0\rvert$	6.45	12.35	14.50

Table 4.2: # (rel.) small singular values of AD^*A^T near optimum; average over 20 instances

4.2 Empirics with Simplex Method

In this section we compare the behaviour of the dual simplex method with instances that have strictly feasible points and instances that do not. We also observe the degeneracy issues that arise in the instances from NETLIB.

4.2.1 Generating Dual LPs without Strict Feasibility

We first show how to generate an instance for the dual feasible set $\mathcal{G}$ that fails strict feasibility. The construction is similar to the one in Section 4.1.1. We generate a degenerate problem by finding a feasible auxiliary system (3.13). Given $m, n, r \in \mathbb{N}$, we construct $A \in \mathbb{R}^{m\times n}$ and $c \in \mathbb{R}^n$ that satisfy (3.13) with $\dim(\text{relint}(\mathcal{G})) = m + r$.

1. Pick any $0 \neq w \in \mathbb{R}^n_+$ with $|\operatorname{supp}(w)| = n - r$. Let

$$\{w\}^\perp = \text{span}\{a_i\}_{i=1}^{n-1} \quad \left(= \text{null}(w^T)\right).$$

   We let the rows of the matrix $A \in \mathbb{R}^{m\times n}$ consist of a random linear combination of the row vectors in the set $\{a_i^T\}_{i=1}^{n-1}$. We note that $Aw = 0$.

2. Pick $s \in \mathbb{R}^n_+$ so that

$$s_i = \begin{cases} 0 & \text{if } i \in \operatorname{supp}(w) \\ \text{positive} & \text{if } i \notin \operatorname{supp}(w). \end{cases}$$

   We note that $\langle w, s\rangle = 0$ holds.

3. Pick $y \in \mathbb{R}^m$ and set $c = A^T y + s$. We note that $\langle c, w\rangle = 0$ holds.

For the empirics, we construct the objective function $b^T y$ of $(\mathcal{D})$ by choosing a vector $\hat{x} \in \mathbb{R}^n_{++}$ and setting $b = A\hat{x}$.

4.2.2 Empirics on the Number of Degenerate Iterations

In this section we test how the lack of strict feasibility affects the performance of the dual simplex method. We choose MOSEK for our tests since MOSEK reports the percentage of degenerate iterations as a part of the solver report. MOSEK reports the quantity 'DEGITER(%)', the ratio of degenerate iterations.

Given a set $\mathcal{G}$ and a point $(y, s) \in \text{relint}(\mathcal{G}) \subseteq \mathbb{R}^m \oplus \mathbb{R}^n_+$, let r be the number of positive entries of s, i.e., $r = |\operatorname{supp}(s)|$. In our tests, we gradually increase r for fixed n, m and generate instances for $\mathcal{G}$ as described in Section 4.2.1. We then observe the behaviour of the dual simplex method. Table 4.3 contains the results. In Table 4.3, a smaller value for the header $(r/n)\%$ means that there are more entries of s that are identically 0 in the set $\mathcal{G}$; and the value 0% means that strict

		$100\% - (r/n)\%$				
		40	30	20	10	0
(n,m)	(1000, 250)	36.62	10.18	0.01	0.02	0.00
	(2000, 500)	39.72	18.28	0.07	0.15	0.01
	(3000, 750)	25.99	10.66	0.32	0.75	0.02
	(4000, 1000)	29.78	18.25	0.25	0.53	0.02

Table 4.3: Average of the ratio of degenerate iterations

feasibility holds. For each triple (n, m, r), we generated 10 instances and we report the average of 'DEGITER(%)' of these instances.

We recall Theorem 3.1: lack of strict feasibility implies that all basic feasible solutions are degenerate. However, we observe more, i.e., from Table 4.3, the frequency of degenerate iterations increases as r decreases. In other words, higher degeneracy of the set $\mathcal{G}$ yields more degenerate iterations when the dual simplex method is used.

4.2.3 NETLIB Problems; Perturbations; Stability

We now illustrate the lack of strict feasibility on instances in the NETLIB data set. We used the following first 20 instances that are in standard form at this link:

25fv47	adlittle	afiro	agg	agg2	agg3	bandm	beaconfd	blend	bnl1
bnl2	brandy	d2q06c	degen2	degen3	e226	fffff800	israel	lotfi	maros_r7

We removed redundant rows to guarantee full row rank of A.

Surprisingly, the Slater condition fails for 14 out of these 20 instances.[13] This has interesting implications for both the interior point and simplex methods. The standard interior point method stopping criteria becomes complicated by the unbounded dual optimal set. For the primal simplex method, every iteration will always visit degenerate **BFS**s. Therefore preprocessing to eliminate the variables fixed at 0 is important. In addition, in order to motivate robust optimization, it is shown in e.g., [3,4] that optimal solutions of many of the NETLIB instances are extremely sensitive to perturbations in the data. We now see this to be the case, and we show that **FR** regularizes the problem and avoids this instability.

We first use the instance degen3 in order to illustrate the consequence of lack of strict feasibility. The data matrix A after removing two redundant rows is 1501-by-2604. After **FR**, we obtain the constraint matrix $P_{\bar{m}}AV$ of size 1226-by-1648. This implies that $2604 - 1648 = 956$ number of variables are identically 0 on the feasible set. Furthermore, $\text{IPS}(\mathcal{F}) = 275$ equality constraints are implicitly redundant. By Item 3 of Corollary 3.11, without **FR**, the degree of degeneracy of every **BFS** is at least 275. Namely, the length of the basis is 1501 and every basis contains at least 275 degenerate indices.

We now illustrate that **FR** gives a more robust model with respect to data perturbations using the instance brandy. Let (A, b) be the data after removing the redundant equalities constraints. Let $(P_{\bar{m}}AV, P_{\bar{m}}b)$ be the data for the facially reduced system. The data matrices A and $P_{\bar{m}}AV$ have the sizes 193-by-303 and 155-by-260, respectively[14]. Set the perturbation scalars $\epsilon_A = \epsilon_b =$

[13] The instances 25fv47, afiro, blend, israel, lotfi and maros_r7 have strictly feasible points.

[14] This also means that, without **FR**, every **BFS** has at least 38 degenerate basic variables. At least 19.69 percent of basic variables are always degenerate.

10^{-9}. We construct a random perturbation matrix Φ, $\|\Phi\|_F = \|A\|_F + 1$, and random perturbation vector ϕ, $\|\phi\|_2 = \|b\|_2 + 1$. We then solve the problem

$$\tilde{p}^* = \max\{\langle c, x\rangle : (A + \epsilon_A \Phi)x = b + \epsilon_b \phi, \ x \geq 0\}.$$

For the facially reduced system, we used the identical perturbation data Φ, ϕ and discard the rows and columns of (A, b) found from **FR**. That is, we use the perturbations $P_{\bar{m}}\Phi V$ and $P_{\bar{m}}\phi$ to the facially reduced system after the scaling $\|P_{\bar{m}}\Phi V\|_F = \|P_{\bar{m}}AV\|_F + 1$ and $\|P_{\bar{m}}\phi\|_2 = \|P_{\bar{m}}b\|_2 + 1$. We then solve

$$\max\{\langle V^T c, v\rangle : (P_{\bar{m}}AV + \epsilon_A P_{\bar{m}}\Phi V)v = P_{\bar{m}}b + \epsilon_b P_{\bar{m}}\phi, \ v \geq 0\}.$$

In this way, we maintain the identical perturbation structure for the original system and the facially reduced system. We also generate a transportation problem and use the aforementioned perturbations. We note that the transportation problems have Slater points but are known to be highly degenerate. The size of the data generated is 49-by-600.

In the experiment, we tested the instances using 100 different perturbation settings. We randomly generated perturbations Φ, ϕ with the density set at 0.1. We used MOSEK simplex with the setting 'MSK_OPTIMIZER_FREE_SIMPLEX'. In Table 4.4, the headers ϵ_A and ϵ_b refer to the scalars used for perturbations as described above. The headers (A, b), $(P_{\bar{m}}AV, P_{\bar{m}}b)$ and $(A_{\text{trans}}, b_{\text{trans}})$ refer to the non-facially reduced system, the facially reduced system and the transportation problems, with the perturbations. The integral values in the table indicate the number of times that the solver outputs PRIMAL_AND_DUAL_FEASIBLE. Let p^* be the optimal value for the unperturbed instance brandy, and let $\tilde{p}^*$ be the optimal value of a perturbed instance of brandy. The non-integral values in the table indicate the average relative difference in the optimal values between p^* and $\tilde{p}^*$. The relative difference is computed using the formula $\frac{|p^* - \tilde{p}^*|}{2|p^* + \tilde{p}^*|}$. For example, the first entry 11 in Table 4.4 means that 100−11 out of 100 perturbed instances yield infeasibility or unknown status, i.e., only 11 solved successfully. The entry 4.938e-02 next to 11 indicates the average of $\frac{|p^* - \tilde{p}^*|}{2|p^* + \tilde{p}^*|}$ on those 11 instances. We see in column (A, b) and the column $(P_{\bar{m}}AV, P_{\bar{m}}b)$ in

ϵ_A	ϵ_b	(A, b)	$(P_{\bar{m}}AV, P_{\bar{m}}b)$	$(A_{\text{trans}}, b_{\text{trans}})$
1.0e-09	0	(11 , 4.938e-02)	(97 , 6.705e-03)	100
0	1.0e-09	(27 , 2.470e-10)	(100 , 2.208e-10)	100
1.0e-09	1.0e-09	(11 , 1.339e-01)	(96 , 8.719e-03)	100

Table 4.4: Number of successful results out of 100 perturbed instances using simplex method on the instance brandy and transportation problem

Table 4.4, that the facially reduced problems are more immune to data perturbations; the number of successfully solved perturbed instances are significantly larger and the optimal values under the perturbations are less influenced. The last column indicates that although the instance may have many degenerate **BFS**s, having a strictly feasible point is important in terms of perturbations in data, i.e., this emphasizes the difference between the two types of degeneracy.

5 Conclusion

We have addressed the impact, for both theoretical and computational reasons, of loss of strict feasibility in **LP**, one type of degeneracy at a **BFS**. For our numerics we illustrated this using the accuracy of optimality conditions as well as the effect of perturbations, for the two most popular classes of algorithms, i.e., simplex and interior point methods. For the theory, we proved, using the two-step facial reduction, that if strict feasibility fails for a linear program, then every **BFS** is degenerate. In addition, we showed that facial reduction can be implemented efficiently to obtain a smaller simpler problem with strict feasibility, and that this improves stability. This was illustrated on random problems, as well as instances from the NETLIB data set.

An essential step for almost all algorithms for linear programming is preprocessing. One part of preprocessing is identifying *fixed variables*. However, identifying variables fixed at 0, facial reduction, has not been done due to expense and accuracy problems. In this paper we have shown that not eliminating these variables, i.e., lack of strict feasibility, is equivalent to implicit singularity and this helps explain the numerical difficulties that arise. We have further provided an efficient preprocessing step for facial reduction, i.e., we continue on phase I of the simplex method that eliminates all the artificial variables, and eliminate the variables fixed at 0. We observed that a variable that is basic (positive) in every **BFS** corresponds to a redundant constraints and, by complementary slackness, corresponds to a variable fixed at 0 in the dual. And redundant constraints have been shown in the literature to poorly affect algorithms [18]. Moreover, identifying redundant constraints is a nontrivial operation e.g., [10]. This motivates doing **FR** on both the primal and the dual problems. (It is still unclear whether or not we have to repeat **FR** on the primal again.)

We have presented various numerical experiments that convey the importance of preprocessing for strict feasibility for linear programs, Section 4. For interior point methods, we illustrated the importance of strict feasibility using condition numbers and relationships with *nearness to infeasibility*. We also shed light on the main difficulties that arose with the implicit redundant constraints and used the QR decomposition to show how these difficulties come into play. This also relates to the implicit problem singularity, IPS. A larger IPS means that there is a higher chance of inducing an infeasible problem under perturbations. A large number of degenerate **BFS**s is believed to cause difficulties for the simplex method. We have shown that the settings for having many identically 0 variables in the dual program yield many degenerate iterations. We also have shown that many NETLIB instances fail strict feasibility and used selected instances to show the effect of this degeneracy. Moreover, the facially reduced problems are seen to be more robust with respect to data perturbations. This further emphasizes that ensuring strict feasibility should be part of preprocessing for linear programming.

Our results can easily extend to other forms of **LP**s and to more general problems where degeneracies arise, such as the active set method for quadratic programs [23, 48]. We are currently extending the efficient **FR** technique to semidefinite programs.

Acknowledgements

This research is supported by the National Sciences and Engineering Research Council (NSERC) of Canada, Grant # No. 50503-10827.

Index

References

[1] E.D. Andersen. Finding all linearly dependent rows in large-scale linear programming. *Optimization methods & software*, 6(3):219–227, 1995. 8

[2] E.D. Andersen and K.D. Andersen. Presolving in linear programming. *Math. Program.*, 71(2):221–245, 1995. 5, 8

[3] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization.* Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2009. 29

[4] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Oper. Res. Lett.*, 25(1):1–13, 1999. 29

[5] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Belmont, MA, 1997. 22

[6] R.E. Bixby. Solving real-world linear programs: a decade and more of progress. *Oper. Res.*, 50(1):3–15, 2002. 50th anniversary issue of Operations Research. 3

[7] R.G. Bland. New finite pivoting rules for the simplex method. *Math. Oper. Res.*, 2(2):103–107, 1977. 3

[8] J.M. Borwein and H. Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Austral. Math. Soc. Ser. A*, 30(3):369–380, 1980/81. 6

[9] J.M. Borwein and H. Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. Appl.*, 83(2):495–530, 1981. 6

[10] R.J. Caron, A. Boneh, and S. Boneh. Redundancy. In *Advances in sensitivity analysis and parametric programming*, volume 6 of *Internat. Ser. Oper. Res. Management Sci.*, pages 13.1–13.41. Kluwer Acad. Publ., Boston, MA, 1997. 31

[11] R. Chandrasekaran, Santosh N. Kabadi, and Katta G. Murty. Some NP-complete problems in linear programming. *Oper. Res. Lett.*, 1(3):101–104, 1981/82. 13

[12] A. Charnes. Optimality and degeneracy in linear programming. *Econometrica*, 20:160–170, 1952. 3

[13] Y.-L. Cheung. *Preprocessing and Reduction for Semidefinite Programming via Facial Reduction: Theory and Practice.* PhD thesis, University of Waterloo, 2013. 22

[14] Y-L. Cheung, S. Schurr, and H. Wolkowicz. Preprocessing and regularization for degenerate semidefinite programs. In D.H. Bailey, H.H. Bauschke, P. Borwein, F. Garvan, M. Thera, J. Vanderwerff, and H. Wolkowicz, editors, *Computational and Analytical Mathematics, In Honor of Jonathan Borwein's 60th Birthday*, volume 50 of *Springer Proceedings in Mathematics & Statistics*, pages 225–276. Springer, 2013. 7

[15] V. Chvátal. *Linear programming.* A Series of Books in the Mathematical Sciences. W. H. Freeman and Company, New York, 1983. 8

[16] G.B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, New Jersey, 1963. 3

[17] G.B. Dantzig, A. Orden, and P. Wolfe. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific J. Math.*, 5:183–195, 1955. 3

[18] A. Deza, E. Nematollahi, R. Peyghami, and T. Terlaky. The central path visits all the vertices of the Klee-Minty cube. *Optim. Methods Softw.*, 21(5):851–865, 2006. 5, 31

[19] A. Deza, E. Nematollahi, and T. Terlaky. How good are interior point methods? Klee-Minty cubes tighten iteration-complexity bounds. *Math. Program.*, 113(1, Ser. A):1–14, 2008. 5

[20] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2, Ser. A):201–213, 2002. 24

[21] D. Drusvyatskiy, G. Li, and H. Wolkowicz. A note on alternating projections for ill-posed semidefinite feasibility problems. *Math. Program.*, 162(1-2, Ser. A):537–548, 2017. 7

[22] D. Drusvyatskiy and H. Wolkowicz. The many faces of degeneracy in conic optimization. *Foundations and Trends® in Optimization*, 3(2):77–170, 2017. 6, 7, 10, 20

[23] A. Forsgren, P.E. Gill, and E. Wong. Primal and dual active-set methods for convex quadratic programming. *Mathematical programming*, 159(1-2):469–508, 2015. 31

[24] R.M. Freund and F. Ordonez. On an extension of condition number theory to nonconic convex optimization. *Mathematics of operations research*, 30(1):173–194, 2005. 18, 20

[25] R.M. Freund and J.R. Vera. Some characterizations and properties of the "distance to ill-posedness" and the condition measure of a conic linear system. Technical report, MIT, Cambridge, MA, 1997. 18, 20

[26] T. Gal, editor. *Degeneracy in optimization problems*. Baltzer Science Publishers BV, Bussum, 1993. Ann. Oper. Res. **46**/47 (1993), no. 1-4. 3

[27] J. Gauvin. A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming. *Mathematical programming*, 12(1):136–138, 1977. 20

[28] A.J. Goldman and A.W. Tucker. Theory of linear programming. In *Linear inequalities and related systems*, pages 53–97. Princeton University Press, Princeton, N.J., 1956. Annals of Mathematics Studies, no. 38. 3, 21

[29] J. Gondzio. Presolve analysis of linear programs prior to applying an interior point method. *INFORMS J. Comput.*, 9(1):73–91, 1997. 8

[30] M. Gonzalez-Lima, H. Wei, and H. Wolkowicz. A stable primal-dual approach for linear programming under nondegeneracy assumptions. *Comput. Optim. Appl.*, 44(2):213–247, 2009. 3

[31] N. Gould and J. Scott. A note on performance profiles for benchmarking software. *ACM transactions on mathematical software*, 43(2):1–5, 2016. 24

[32] O. Güler, D. Den Hertog, C. Roos, T. Terlaky, and T. Tsuchiya. Degeneracy in interior point methods for linear programming: a survey. *Ann. Oper. Res.*, 46/47(1-4):107–138, 1993. Degeneracy in optimization problems. 3, 23, 27

[33] X. Huang. Preprocessing and postprocessing in linear optimization. Master's thesis, McMaster University, 2004. 5, 8

[34] J. Im and H. Wolkowicz. A strengthened Barvinok-Pataki bound on SDP rank. *Oper. Res. Lett.*, 49(6):837–841, 2021. 11 pages, accepted Aug. 2021. 6, 7, 10

[35] O. L. Mangasarian and S. Fromovitz. The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. *J. Math. Anal. Appl.*, 17:37–47, 1967. 3

[36] N. Megiddo. A note on degeneracy in linear programming. *Math. Programming*, 35(3):365–367, 1986. 3

[37] C. Mészáros and U.H. Suhl. Advanced preprocessing techniques for linear and quadratic programming. *OR Spectrum*, 25:575–595, 2003. 10.1007/s00291-003-0130-x. 8

[38] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Math. Oper. Res.*, 23(2):339–358, 1998. 9

[39] F.N. Permenter. *Reduction methods in semidefinite and conic optimization*. PhD thesis, Massachusetts Institute of Technology, 2017. 6, 10

[40] D.W. Peterson. A review of constraint qualifications in finite-dimensional spaces. *SIAM Rev.*, 15:639–654, 1973. 11

[41] J. Renegar. Some perturbation theory for linear programming. *Math. Programming*, 65(1, Ser. A):73–91, 1994. 18

[42] J. Renegar. Incorporating condition measures into the complexity theory of linear programming. *SIAM J. Optim.*, 5(3):506–524, 1995. 20

[43] D. M. Ryan and M. R. Osborne. On the solution of highly degenerate linear programmes. *Math. Program.*, 41:385–392, 1988. 3

[44] S. Sremac. *Error bounds and singularity degree in semidefinite programming*. PhD thesis, University of Waterloo, 2019. 6, 7, 10

[45] S. Sremac, H.J. Woerdeman, and H. Wolkowicz. Error bounds and singularity degree in semidefinite programming. *SIAM J. Optim.*, 31(1):812–836, 2021. 7

[46] J.F. Sturm. Error bounds for linear matrix inequalities. *SIAM J. Optim.*, 10(4):1228–1248 (electronic), 2000. 7

[47] T. Terlaky and S.Z. Zhang. Pivot rules for linear programming: a survey on recent theoretical developments. *Ann. Oper. Res.*, 46/47(1-4):203–233, 1993. Degeneracy in optimization problems. 3

[48] P. Wolfe. The simplex method for quadratic programming. *Econometrica*, 27(3):382–398, 1959. 31

[49] S. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, 1996. 17, 20, 22