Math. Program., Ser. A 88: 451–485 (2000)

Digital Object Identifier (DOI) 10.1007/s101070000165

Marina Epelman* · Robert M. Freund

Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system

Received: March 13, 1997 / Accepted: March 9, 2000
Published online July 20, 2000 – © Springer-Verlag 2000

Abstract. A conic linear system is a system of the form

$$\begin{array}{ll}(\mathrm{FP}_d) & Ax = b \\ & x \in C_X,\end{array}$$

where $A : X \longrightarrow Y$ is a linear operator between n- and m-dimensional linear spaces X and Y, $b \in Y$, and $C_X \subset X$ is a closed convex cone. The data for the system is $d = (A, b)$. This system is "well-posed" to the extent that (small) changes in the data $d = (A, b)$ do not alter the status of the system (the system remains feasible or not). Renegar defined the "distance to ill-posedness," $\rho(d)$, to be the smallest change in the data $\Delta d = (\Delta A, \Delta b)$ needed to create a data instance $d + \Delta d$ that is "ill-posed," i.e., that lies in the intersection of the closures of the sets of feasible and infeasible instances $d' = (A', b')$ of $(\mathrm{FP}_{(\cdot)})$. Renegar also defined the condition number $\mathcal{C}(d)$ of the data instance d as the scale-invariant reciprocal of $\rho(d)$: $\mathcal{C}(d) \stackrel{\triangle}{=} \frac{\|d\|}{\rho(d)}$.

In this paper we develop an elementary algorithm that computes a solution of (FP_d) when it is feasible, or demonstrates that (FP_d) has no solution by computing a solution of the alternative system. The algorithm is based on a generalization of von Neumann's algorithm for solving linear inequalities. The number of iterations of the algorithm is essentially bounded by

$$O\big(\tilde{c}\,\mathcal{C}(d)^2 \ln(\mathcal{C}(d))\big)$$

where the constant $\tilde{c}$ depends only on the properties of the cone C_X and is independent of data d. Each iteration of the algorithm performs a small number of matrix-vector and vector-vector multiplications (that take full advantage of the sparsity of the original data) plus a small number of other operations involving the cone C_X. The algorithm is "elementary" in the sense that it performs only a few relatively simple computations at each iteration.

The solution $\hat{x}$ of the system (FP_d) generated by the algorithm has the property of being "reliable" in the sense that the distance from $\hat{x}$ to the boundary of the cone C_X, $\mathrm{dist}(\hat{x}, \partial C_X)$, and the size of the solution, $\|\hat{x}\|$, satisfy the following inequalities:

$$\|\hat{x}\| \le c_1 \mathcal{C}(d),\ \mathrm{dist}(\hat{x}, \partial C_X) \ge c_2 \frac{1}{\mathcal{C}(d)},\ \text{and}\ \frac{\|\hat{x}\|}{\mathrm{dist}(\hat{x}, \partial C_X)} \le c_3 \mathcal{C}(d),$$

where c_1, c_2, c_3 are constants that depend only on properties of the cone C_X and are independent of the data d (with analogous results for the alternative system when the system (FP_d) is infeasible).

Key words. complexity of convex programming – conditioning – error analysis

M. Epelman: University of Michigan, Industrial and Operations Engineering, 1205 Beal Avenue, Ann Arbor, MI 48109-2117, USA, e-mail: mepelman@umich.edu

R.M. Freund: M.I.T. Sloan School of Management, 50 Memorial Drive, Cambridge, MA 02142-1347, USA, e-mail: rfreund@mit.edu

Mathematics Subject Classification (1991): 90C, 90C05, 90C60

* This research has been partially supported through an NSF Graduate Research Fellowship

1. Introduction

The subject of this paper is the development of an algorithm for solving a convex feasibility problem in conic linear form:

$$\begin{array}{rl} (\text{FP}_d) & Ax = b \\ & x \in C_X, \end{array} \tag{1}$$

where $A : X \longrightarrow Y$ is a linear operator between the (finite) n-dimensional normed linear vector space X and the (finite) m-dimensional normed linear vector space Y (with norms $\|x\|$ for $x \in X$ and $\|y\|$ for $y \in Y$, respectively), $C_X \subset X$ is a closed convex cone, and $b \in Y$. We denote by $d = (A, b)$ the "data" for the problem (FP_d). That is, the cone C_X is regarded as fixed and given, and the data for the problem is the linear operator A together with the vector b. We denote the set of solutions of (FP_d) as X_d to emphasize the dependence on the data d, i.e.,

$$X_d = \{x \in X : Ax = b,\ x \in C_X\}.$$

The problem (FP_d) is a very general format for studying the feasible regions of convex optimization problems, and has recently received much attention in the analysis of interior-point methods, see Nesterov and Nemirovskii [22] and Renegar [29] and [30], among others, wherein interior-point methods for (FP_d) are shown to be theoretically efficient.

We develop an algorithm called "algorithm CLS" (for Conic Linear System) that either computes a solution of the system (FP_d), or demonstrates that (FP_d) is infeasible by computing a solution of an alternative (dual) system. In both cases the solution provided by algorithm CLS is "reliable" in a sense that will be described shortly.

Algorithm CLS is based on a generalization of the algorithm privately communicated by von Neumann to Dantzig and studied by Dantzig in [6] and [7], and is part of a large class of "elementary" algorithms for finding a point in a suitably described convex set, such as reflection algorithms for linear inequality systems (see [1,21,8,15]), the perceptron algorithm [31–34], and other so-called row-action methods. When applied to linear inequality systems, these elementary algorithms share the following desirable properties, namely: the work per iteration is extremely low (typically involving only a few matrix-vector or vector-vector multiplications), and the algorithms fully exploit the sparsity of the original data at each iteration. We refer to these algorithms as "elementary" in that the algorithms do not perform particularly sophisticated computations at each iteration, and in some sense these algorithms are all very unsophisticated as a result (especially compared to an interior-point algorithm or a volume-reducing cutting-plane algorithm such as the ellipsoid algorithm).

In analyzing the complexity of algorithm CLS, we adopt the relatively new concept of the *condition number* $\mathcal{C}(d)$ of (FP_d) developed by Renegar in a series of papers [28–30]. $\mathcal{C}(d)$ is essentially a scale invariant reciprocal of the smallest data perturbation $\Delta d = (\Delta A, \Delta b)$ for which the system ($\text{FP}_{d+\Delta d}$) changes its feasibility status. The problem (FP_d) is well-conditioned to the extent that $\mathcal{C}(d)$ is small; when the problem (FP_d) is "ill-posed" (i.e., arbitrarily small perturbations of the data can yield both feasible and infeasible problem instances), then $\mathcal{C}(d) = +\infty$. The condition number $\mathcal{C}(d)$

is connected to sizes of solutions and deformations of X_d under data perturbations [28], certain geometric properties of X_d [13], and the complexity of algorithms for computing solutions of (FP$_d$) [30, 14]. (The concepts underlying $\mathcal{C}(d)$ will be reviewed in detail at the end of this section.) We show in Sect. 5 that algorithm CLS will compute a feasible solution of (FP$_d$) in

$$O\big(\tilde{c}_1\mathcal{C}(d)^2\ln(\mathcal{C}(d))\big) \tag{2}$$

iterations when (FP$_d$) is feasible, or will demonstrate infeasibility in

$$O\big(\tilde{c}_2\mathcal{C}(d)^2\big) \tag{3}$$

iterations when (FP$_d$) is infeasible. The scalar quantities $\tilde{c}_1$ and $\tilde{c}_2$ are constants that depend only on the simple notion of the "width" of the cones C_X and C_X^* and are independent of the data d, but may depend on the dimension n.

As alluded to above, algorithm CLS will compute a reliable solution of the system (FP$_d$), or will demonstrate that (FP$_d$) is infeasible by computing a reliable solution of an alternative system. We consider a solution $\hat{x}$ of the system (FP$_d$) to be reliable if, roughly speaking, (i) the distance from $\hat{x}$ to the boundary of the cone C_X, dist$(\hat{x}, \partial C_X)$, is not excessively small, (ii) the norm of the solution $\|\hat{x}\|$ is not excessively large, and (iii) the ratio $\frac{\|\hat{x}\|}{\text{dist}(\hat{x},\partial C_X)}$ is not excessively large. A reliable solution of the alternative system is defined similarly. The sense of what is meant by "excessive" is measured using the condition number $\mathcal{C}(d)$. The importance of computing a reliable solution can be motivated by considerations of finite-precision computations. Suppose, for example, that a solution $\hat{x}$ of the problem (FP$_d$) (computed as an output of an algorithm involving iterates $x^1, \ldots, x^k = \hat{x}$, and/or used as input to another algorithm) has the property that dist$(\hat{x}, \partial C_X)$ is very small. Then the numerical precision requirements for checking or guaranteeing feasibility of iterates will necessarily be large. Similar remarks hold for the case when $\|\hat{x}\|$ and/or the ratio $\frac{\|\hat{x}\|}{\text{dist}(\hat{x},\partial C_X)}$ is very large.

In [13] it is shown that when the system (FP$_d$) is feasible, there exists a point $\tilde{x} \in X_d$ such that

$$\|\tilde{x}\| \le c_1\mathcal{C}(d),\ \text{dist}(\tilde{x}, \partial C_X) \ge c_2\frac{1}{\mathcal{C}(d)},\ \text{and}\ \frac{\|\tilde{x}\|}{\text{dist}(\tilde{x}, \partial C_X)} \le c_3\mathcal{C}(d), \tag{4}$$

where the scalar quantities c_1, c_2, and c_3 depend only on the width of the cone C_X, and are independent of the data d of the problem (FP$_d$), but may depend on the dimension n. Algorithm CLS will compute a solution $\hat{x}$ with bounds of the same order as (4), which justifies the term "reliable" solution. Similar remarks hold for the case when (FP$_d$) is infeasible.

It is interesting to compare the complexity bounds of algorithm CLS in (2) and (3) to that of other algorithms for solving (FP$_d$). In [30], Renegar presented an interior-point (i.e., barrier) algorithm for resolving (FP$_d$) and analyzed its performance in terms of the barrier parameter for the cone C_X, and $\mathcal{C}(d)$. In [14] several efficient volume-reducing cutting-plane algorithms for resolving (FP$_d$) (such as the ellipsoid algorithm) are analyzed in terms of $\mathcal{C}(d)$. Both the interior-point algorithm and the ellipsoid algorithm have an iteration complexity bound that is linear in $\ln(\mathcal{C}(d))$, and so are efficient algorithms in a sense defined by Renegar [29].

In contrast with the above efficient algorithms, algorithm CLS developed in this paper has iteration complexity *exponential* in $\ln(\mathcal{C}(d))$. On the other hand, both the interior-point algorithm and the ellipsoid algorithm are very sophisticated algorithms and require significant computational effort to perform each iteration, unlike the elementary algorithm CLS. The interior-point algorithm makes implicit and explicit use of information from a self-concordant barrier at each iteration, and uses this information in the computation of the next iterate by solving for the Newton step along the central trajectory. The work per iteration is $O(n^3)$ operations to compute the Newton step. The ellipsoid algorithm makes use of a separation oracle for the cone C_X in order to perform a special space dilation at each iteration, and the work per iteration of the ellipsoid algorithm is $O(n^2)$ operations. Intuition strongly suggests that the sophistication of these methods is responsible for their excellent computational complexity. In contrast, the elementary algorithm CLS relies only on relatively simple assumptions regarding the ability to work conveniently with the cone C_X (discussed in detail in Sect. 2) and does not perform any sophisticated computation at each iteration. As a result, the work per iteration of algorithm CLS is low, and each iteration fully exploits the sparsity of the original data.

The results in this paper provide positive answers to the following two theoretical questions:

- Is there an elementary algorithm which obtains reliable solutions of well-posed instances of (FP$_d$)?
- Can the iteration complexity of an elementary algorithm for (FP$_d$) be bounded in terms of the condition number $\mathcal{C}(d)$?

This paper does not attempt to address the *practical* performance of algorithm CLS versus theoretically efficient algorithms such as interior-point algorithms or the ellipsoid algorithm. However, we briefly discuss computational performance of a family of algorithms related to CLS in Sect. 6.

An outline of the paper is as follows. The remainder of this introductory section discusses the condition number $\mathcal{C}(d)$ of the system (FP$_d$). Section 2 contains further notation, definitions, assumptions, and preliminary results. Section 3 presents a generalization of the von Neumann algorithm (appropriately called algorithm GVNA) that can be applied to conic linear systems in a special compact form (i.e., with a compactness constraint added). We analyze the properties of the iterates of algorithm GVNA under different termination criteria in Lemmas 1, 2 and 3. Section 4 presents the development of algorithms HCI (Homogeneous Conic Inequalities) and HCE (Homogeneous Conic Equalities) for resolving two essential types of homogeneous conic linear systems. Both algorithms HCI and HCE consist of calls to algorithm GVNA applied to appropriate transformations of the homogeneous systems at hand. Finally, in Sect. 5, we present algorithm CLS for the conic linear system (FP$_d$). Algorithm CLS is a combination of algorithms HCI and HCE. Theorem 3 contains the main complexity result for algorithm CLS, and is the main result of this paper. Section 6 contains some discussion.

We now present the development of the concepts of condition numbers and data perturbation for (FP$_d$) in detail. Recall that $d = (A, b)$ is the data for the problem (FP$_d$). The space of all data $d = (A, b)$ for (FP$_d$) is denoted by $\mathcal{D}$:

$$\mathcal{D} = \{d = (A, b) : A \in L(X, Y), b \in Y\}.$$

For $d = (A, b) \in \mathcal{D}$ we define the product norm on the Cartesian product $L(X, Y) \times Y$ to be

$$\|d\| = \|(A, b)\| = \max\{\|A\|, \|b\|\} \tag{5}$$

where $\|b\|$ is the norm specified for Y and $\|A\|$ is the operator norm, namely

$$\|A\| = \max\{\|Ax\| : \|x\| \leq 1\}. \tag{6}$$

We define

$$\mathcal{F} = \{(A, b) \in \mathcal{D} : \text{ there exists } x \text{ satisfying } Ax = b,\ x \in C_X\}, \tag{7}$$

the set of data instances d for which (FP$_d$) is feasible. Its complement is denoted by $\mathcal{F}^{\mathcal{C}}$, the set of data instances for which (FP$_d$) is infeasible.

The boundary of $\mathcal{F}$ and of $\mathcal{F}^{\mathcal{C}}$ is precisely the set $\mathcal{B} = \partial\mathcal{F} = \partial\mathcal{F}^{\mathcal{C}} = \text{cl}(\mathcal{F}) \cap \text{cl}(\mathcal{F}^{\mathcal{C}})$, where ∂S denotes the boundary and cl(S) denotes the closure of a set S. Note that if $d = (A, b) \in \mathcal{B}$, then (FP$_d$) is ill-posed in the sense that arbitrarily small changes in the data $d = (A, b)$ can yield instances of (FP$_d$) that are feasible, as well as instances of (FP$_d$) that are infeasible. Also, note that $\mathcal{B} \neq \emptyset$, since $d = (0, 0) \in \mathcal{B}$.

For a data instance $d = (A, b) \in \mathcal{D}$, the *distance to ill-posedness* is defined to be:

$$\rho(d) \stackrel{\triangle}{=} \inf\{\|\Delta d\| : d + \Delta d \in \mathcal{B}\} = \begin{cases} \inf\{\|d - \bar{d}\| : \bar{d} \in \mathcal{F}^C\} \text{ if } d \in \mathcal{F}, \\ \inf\{\|d - \bar{d}\| : \bar{d} \in \mathcal{F}\} \quad \text{if } d \in \mathcal{F}^C, \end{cases} \tag{8}$$

see Renegar [28–30]. The *condition number* $\mathcal{C}(d)$ of the data instance d is defined to be:

$$\mathcal{C}(d) = \frac{\|d\|}{\rho(d)} \tag{9}$$

when $\rho(d) > 0$, and $\mathcal{C}(d) = \infty$ when $\rho(d) = 0$. The condition number $\mathcal{C}(d)$ is a measure of the relative conditioning of the data instance d, and can be viewed as a scale-invariant reciprocal of $\rho(d)$, as it is elementary to demonstrate that $\mathcal{C}(d) = \mathcal{C}(\alpha d)$ for any positive scalar α. Observe that since $\tilde{d} = (\tilde{A}, \tilde{b}) = (0, 0) \in \mathcal{B}$, then for any $d \notin \mathcal{B}$ we have $\|d\| = \|d - \tilde{d}\| \geq \rho(d)$, whereby $\mathcal{C}(d) \geq 1$. Further analysis of the distance to ill-posedness has been presented in [13], Vera [35,36,38,37], Filipowski [11,12], Nunez and Freund [23], Peña [25,24] and Peña and Renegar [26].

2. Preliminaries, assumptions, and further notation

We will work in the setup of finite dimensional normed linear vector spaces. Both X and Y are normed linear spaces of finite dimension n and m, respectively, endowed with norms $\|x\|$ for $x \in X$ and $\|y\|$ for $y \in Y$. For $\bar{x} \in X$, let $B(\bar{x}, r)$ denote the ball centered at $\bar{x}$ with radius r, i.e.,

$$B(\bar{x}, r) = \{x \in X : \|x - \bar{x}\| \leq r\},$$

and define $B(\bar{y}, r)$ analogously for $\bar{y} \in Y$.

We denote the set of real numbers by $\Re$ and the set of nonnegative real numbers by $\Re_+$.

We associate with X and Y the dual spaces X^* and Y^* of linear functionals defined on X and Y, respectively, and whose (dual) norms are denoted by $\|u\|_*$ for $u \in X^*$ and $\|w\|_*$ for $w \in Y^*$. Let $c \in X^*$. In order to maintain consistency with standard linear algebra notation in mathematical programming, we will consider c to be a column vector in the space X^* and will denote the linear function $c(x)$ by $c^t x$. Similarly, for $A \in L(X, Y)$ and $f \in Y^*$, we denote $A(x)$ by Ax and $f(y)$ by $f^t y$. We denote the adjoint of A by A^t.

We now recall some facts about norms. Given a finite dimensional linear vector space X endowed with a norm $\|x\|$ for $x \in X$, the dual norm induced on the space X^* is denoted by $\|z\|_*$ for $z \in X^*$, and is defined as:

$$\|z\|_* = \max\{z^t x : \|x\| \leq 1\}, \tag{10}$$

and the Hölder inequality $z^t x \leq \|z\|_* \|x\|$ follows easily from this definition. We also point out that if $A = uv^t$, then it is easy to derive that $\|A\| = \|v\|_* \|u\|$.

If C is a convex cone in X, C^* will denote the dual convex cone defined by

$$C^* = \{z \in X^* : z^t x \geq 0 \text{ for any } x \in C\}.$$

We will say that a cone C is *regular* if C is a closed convex cone, has a nonempty interior, and is pointed (i.e., contains no line).

Remark 1. If C is a closed convex cone, then C is regular if and only if C^* is regular.

The "strong alternative" system of (FP$_d$) is:

$$\begin{array}{ll} (\text{SA}_d) & A^t s \in C_X^* \\ & b^t s < 0. \end{array} \tag{11}$$

A separating hyperplane argument yields the following partial theorem of the alternative regarding the feasibility of the system (FP$_d$):

Proposition 1. *If (SA$_d$) is feasible, then (FP$_d$) is infeasible. If (FP$_d$) is infeasible, then the following "weak alternative" system (12) is feasible:*

$$\begin{array}{l} A^t s \in C_X^* \\ b^t s \leq 0 \\ s \neq 0. \end{array} \tag{12}$$

When the system (FP$_d$) is well-posed, we have the following strong theorem of the alternative:

Proposition 2. *Suppose $\rho(d) > 0$. Then exactly one of the systems (FP$_d$) and (SA$_d$) is feasible.*

We denote the set of solutions of (SA_d) as A_d, i.e.,

$$A_d = \{s \in Y^* : A^t s \in C_X^*,\ b^t s < 0\}.$$

Similarly to solutions of (FP_d), we consider a solution $\hat{s}$ of the system (SA_d) to be reliable if the ratio $\frac{\|\hat{s}\|_*}{\mathrm{dist}(\hat{s}, \partial A_d)}$ is not excessively large. (Because the system (SA_d) is homogeneous, it makes little sense to bound $\|\hat{s}\|_*$ from above or to bound $\mathrm{dist}(\hat{s}, \partial A_d)$ from below, as all solutions can be scaled by any positive quantity.) In [13] it is shown that when the system (FP_d) is infeasible, there exists a point $\tilde{s} \in A_d$ such that

$$\frac{\|\tilde{s}\|_*}{\mathrm{dist}(\tilde{s}, \partial A_d)} \leq c_4 \mathcal{C}(d), \tag{13}$$

where the scalar quantity c_4 depends only on the width of the cone C_X^*. (The concept of the width of a cone will be defined in the next paragraph.) Algorithm CLS will compute a solution $\hat{s}$ with a bound of the same order as (13).

Let C be a regular cone in the normed linear vector space X. We will use the following definition of the *width* of C:

Definition 1. *If C is a regular cone in the normed linear vector space X, the width of C is given by:*

$$\tau_C = \max\left\{\frac{r}{\|x\|} : B(x, r) \subset C\right\}.$$

We remark that τ_C measures the maximum ratio of the radius to the norm of the center of an inscribed ball in C, and so larger values of τ_C correspond to an intuitive notion of greater width of C. Note that $\tau_C \in (0, 1]$, since C has a nonempty interior and C is pointed, and τ_C is attained for some $(\bar{x}, \bar{r})$ as well as along the ray $(\alpha\bar{x}, \alpha\bar{r})$ for all $\alpha > 0$. By choosing the value of α appropriately, we can find $u \in C$ such that $\|u\| = 1$ and τ_C is attained for (u, τ_C).

Closely related to the width is the notion of the *coefficient of linearity* for a cone C:

Definition 2. *If C is a regular cone in the normed linear vector space X, the coefficient of linearity for the cone C is given by:*

$$\begin{array}{lll} \beta_C = \sup & & \inf u^t x \\ & u \in X^* & x \in C \\ & \|u\|_* = 1 & \|x\| = 1. \end{array} \tag{14}$$

The coefficient of linearity β_C measures the extent to which the norm $\|x\|$ can be approximated by a linear function over the cone C. We have the following properties of β_C:

Remark 2 (see [13]). $0 < \beta_C \leq 1$. There exists $\bar{u} \in \mathrm{int}\, C^*$ such that $\|\bar{u}\|_* = 1$ and $\beta_C = \min\{\bar{u}^t x : x \in C,\ \|x\| = 1\}$. For any $x \in C$, $\beta_C \|x\| \leq \bar{u}^t x \leq \|x\|$. The set $\{x \in C : \bar{u}^t x = 1\}$ is a bounded and closed convex set.

In light of Remark 2 we refer to $\bar{u}$ as the norm linearization vector for the cone C. The following proposition provides insight into the relationship between the width of C and the coefficient of linearity for C^*:

Proposition 3 (see [14]). *Suppose that C is a regular cone in the normed linear vector space X, and let τ_C denote the width of C and let β_{C^*} denote the coefficient of linearity for C^*. Then $\tau_C = \beta_{C^*}$. Moreover, τ_C is attained for (u, τ_C), where u is the norm linearization vector for the cone C^*.*

We now pause to illustrate the above notions on two relevant instances of the cone C, namely the nonnegative orthant $\Re^n_+$ and the positive semi-definite cone $S^{k\times k}_+$. We first consider the nonnegative orthant. Let $X = \Re^n$ and $C = \Re^n_+ \stackrel{\triangle}{=} \{x \in \Re^n : x \geq 0\}$. Then we can identify X^* with X and in so doing, $C^* = \Re^n_+$ as well. If $\|x\|$ is given by the L_1 norm $\|x\| = \sum_{j=1}^n |x_j|$, then note that $\|x\| = e^t x$ for all $x \in C$ (where e is the vector of ones), whereby the coefficient of linearity is $\beta_C = 1$ and $\bar{u} = e$. If instead of the L_1 norm, the norm $\|x\|$ is the L_p norm defined by:

$$\|x\|_p = \left(\sum_{j=1}^n |x_j|^p \right)^{1/p}$$

for $p \geq 1$, then it is straightforward to show that $\bar{u} = \left(n^{\left(\frac{1}{p}-1\right)}\right)e$ and the coefficient of linearity is $\beta_C = n^{\left(\frac{1}{p}-1\right)}$. Also, by setting $u = e$, it is straightforward to show that the width is $\tau_C = n^{-\frac{1}{p}}$.

Now consider the positive semi-definite cone, which has been shown to be of enormous importance in mathematical programming (see Alizadeh [2] and Nesterov and Nemirovskii [22]). Let $X = S^{k\times k}$ denote the set of real $k \times k$ symmetric matrices, and so $n = \frac{k(k+1)}{2}$, and let $C = S^{k\times k}_+ \stackrel{\triangle}{=} \left\{x \in S^{k\times k} : x \succeq 0\right\}$, where $x \succeq 0$ is the Löwner partial ordering, i.e., $x \succeq w$ if $x - w$ is a positive semi-definite symmetric matrix. Then C is a closed convex cone. We can identify X^* with X, and in so doing it is elementary to derive that $C^* = S^{k\times k}_+$. For $x \in X$, let $\lambda(x)$ denote the k-vector of ordered eigenvalues of x. For any $p \geq 1$, let the norm of x be defined by

$$\|x\| = \|x\|_p = \left(\sum_{j=1}^k |\lambda_j(x)|^p \right)^{\frac{1}{p}}$$

(see [19], for example, for a proof that $\|x\|_p$ is a norm). When $p = 1$, $\|x\|_1$ is the sum of the absolute values of the eigenvalues of x. Therefore, when $x \in C$, $\|x\|_1 = tr(x) = \sum_{i=1}^k x_{ii}$ where x_{ij} is the ijth entry of the real matrix x (and $tr(x)$ is the trace of x), and so $\|x\|_1$ is a linear function on C. Therefore, when $p = 1$, we have $\bar{u} = I$ and the coefficient of linearity is $\beta_C = 1$. When $p > 1$, it is easy to show that $\bar{u} = \left(k^{\left(\frac{1}{p}-1\right)}\right)I$ has $\|\bar{u}\|_* = \|\bar{u}\|_q = 1$ (where $1/p + 1/q = 1$) and that $\beta_C = k^{\left(\frac{1}{p}-1\right)}$. Also, it is easy to show by setting $u = I$ that the width is $\tau_C = k^{-\frac{1}{p}}$.

We will make the following assumption throughout the paper concerning the cone C_X and the norm on the space Y:

Assumption 1. *$C_X \subset X$ is a regular cone. The coefficient of linearity β for the cone C_X, and the width τ of the cone C_X, together with corresponding norm linearization vectors $\bar{f}$ (for the cone C_X) and f (for the cone C_X^*) are known and given. For $y \in Y$, $\|y\| = \|y\|_2$.*

Suppose C is a regular cone in the normed vector space X, and $\bar{u}$ is the norm linearization vector for C. Given any linear function $c^t x$ defined on $x \in X$, we define the following conic section optimization problem:

$$\begin{aligned} (\mathrm{CSOP}_C)\ \min_x\ & c^t x \\ \text{s.t. } & x \in C \\ & \bar{u}^t x = 1. \end{aligned} \tag{15}$$

For the algorithm CLS developed in this paper, we presume that we have available an oracle that can solve (CSOP_{C_X}) efficiently, that is, the upper bound on the number of operations the oracle takes to solve (CSOP_{C_X}) is not excessive, for otherwise the algorithm will not be very efficient. Let T_C denote an upper bound on the number of operations performed in a call to the oracle.

We now pause to illustrate how an oracle for solving (CSOP_C) is easily implemented for two relevant instances of the cone C, namely $\Re^n_+$ and $S^{k\times k}_+$. We first consider $\Re^n_+$. As discussed above, when $\|x\|$ is given by L_p norm with $p \geq 1$, the norm approximation vector $\bar{u}$ is a positive multiple of the vector e. Therefore, for any c, the problem (CSOP_C) is simply the problem of finding the index of the smallest element of the vector c, so that the solution of (CSOP_C) is easily computed as $x_c = e^i$, where $i \in \mathrm{argmin}\{c_j : j = 1, \ldots, n\}$. Thus $T_C = n$.

We now consider $S^{k\times k}_+$. As discussed above, when $\|x\|$ is given by $\|x\| = \|x\|_p = \left(\sum_{j=1}^n |\lambda_j(x)|^p\right)^{\frac{1}{p}}$ with $p \geq 1$, the norm approximation vector $\bar{u}$ is a positive multiple of the matrix I. For any $c \in S^{k\times k}$, the problem (CSOP_C) corresponds to the problem of finding the normalized eigenvector corresponding to the smallest eigenvalue of the matrix c, i.e., (CSOP_C) is a minimum eigenvalue problem and is solvable to within machine tolerance in $O(k^3)$ operations in practice (though not in theory).

Solving (CSOP) for the Cartesian product of two cones is easy if (CSOP) is easy to solve for each of the two cones: suppose that $X = V \times W$ with norm $\|x\| = \|(v, w)\| \stackrel{\triangle}{=} \|v\| + \|w\|$, and $C = C_V \times C_W$ where $C_V \subset V$ and $C_W \subset W$ are regular cones with norm linearization vectors $\bar{u}_V$ and $\bar{u}_W$, respectively. Then the norm linearization vector for the cone C is $\bar{u} = (\bar{u}_V, \bar{u}_W)$, $\beta_C = \min\{\beta_{C_V}, \beta_{C_W}\}$, and $T_C = T_{C_V} + T_{C_W} + O(1)$.

We end this section with the following remark which gives a geometric interpretation of the distance from a given point to the boundary of a closed convex set, which will be often used in this paper.

Remark 3. Let S be a closed convex set in $\Re^m$ and let $f \in \Re^m$ be given. The distance from f to the boundary of S is denoted as

$$\mathrm{dist}(f, \partial S) \stackrel{\triangle}{=} \min_z\{\|f - z\| : z \in \partial S\}. \tag{16}$$

If $f \notin S$, then $\mathrm{dist}(f, \partial S) = \min\{\|f - z\| : z \in S\}$. If $f \in S$, then $\mathrm{dist}(f, \partial S) = \max\{r : B(f, r) \subset S\}$.

3. A generalized von Neumann algorithm for a conic linear system in compact form

In this section we consider a generalization of the algorithm of von Neumann studied by Dantzig in [6] and [7], see also [10]. We will work with a conic linear system of the form:

$$\begin{array}{rl} \text{(P)} & Mx = g \\ & x \in C \\ & \bar{u}^t x = 1, \end{array} \tag{17}$$

where $C \subset X$ is a closed convex cone in the (finite) n-dimensional normed linear vector space X, and $g \in Y$ where Y is the (finite) m-dimensional linear vector space with Euclidean norm $\|y\| = \|y\|_2$, and $M \in L(X, Y)$. We assume that C is a regular cone, and the norm linearization vector $\bar{u}$ of Remark 2 is known and given. (The original algorithm of von Neumann presented and analyzed by Dantzig in [6] and [7] was developed for the case when $C = \Re^n_+$ and $\bar{u} = e$.) We will refer to a system of the form (17) as a conic linear system in compact form, or simply a compact-form system.

The "alternative" system to (P) of (17) is:

$$\text{(A)}\ M^t s - \bar{u}(g^t s) \in \operatorname{int} C^*, \tag{18}$$

and a generalization of Farkas' Lemma yields the following duality result:

Proposition 4. *Exactly one of the systems (P) of (17) and (A) of (18) has a solution.*

Notice that the feasibility problem (P) is equivalent to the following optimization problem:

$$\text{(OP)}\ \min_x \{\|g - Mx\| : x \in C,\ \bar{u}^t x = 1\}.$$

If (P) has a feasible solution, the optimal value of (OP) is 0; otherwise, the optimal value of (OP) is strictly positive. We will say that a point x is "admissible" if it is a feasible point for (OP), i.e., $x \in C$ and $\bar{u}^t x = 1$.

We now describe a generic iteration of our algorithm. At the beginning of the iteration we have an admissible point $\bar{x}$. Let $\bar{v}$ be the "residual" at the point $\bar{x}$, namely, $\bar{v} = g - M\bar{x}$. Notice that $\|\bar{v}\| = \|g - M\bar{x}\|$ is the objective value of (OP). The algorithm calls an oracle to solve the following instance of the conic section optimization problem (CSOP_C) of (15):

$$\begin{array}{lll} \min\ \bar{v}^t(g - Mp) = \min\ \bar{v}^t(g\bar{u}^t - M)p \\ p & & p \\ \text{s.t.}\ \ p \in C & & \text{s.t.}\ \ p \in C \\ \quad \bar{u}^t p = 1 & & \quad \bar{u}^t p = 1, \end{array} \tag{19}$$

where (19) is an instance of the (CSOP_C) with $c = (-M^t + \bar{u}g^t)\bar{v}$. Let $\bar{p}$ be an optimal solution to the problem (19), and $\bar{w} = g - M\bar{p}$.

Next, the algorithm checks whether the termination criterion is satisfied. The termination criterion for the algorithm is given in the form of a function $\text{STOP}(\cdot)$, which evaluates to 1 exactly when its inputs satisfy some termination criterion (some relevant

examples are presented after the statement of the algorithm). If $STOP(\cdot) = 1$, the algorithm concludes that the appropriate termination criterion is satisfied and stops.

On the other hand, if $STOP(\cdot) = 0$, the algorithm continues the iteration. The direction $\bar{p} - \bar{x}$ turns out to be a direction of potential improvement of the objective function of (OP). The algorithm takes a step in the direction $\bar{p} - \bar{x}$ with step-size found by constrained line-search. In particular, let

$$\tilde{x}(\lambda) = \bar{x} + \lambda(\bar{p} - \bar{x}),\ \lambda \in [0, 1].$$

Then the next iterate $\tilde{x}$ is computed as $\tilde{x} = \tilde{x}(\lambda^*)$, where λ^* is chosen to minimize the size of the residual at $\tilde{x}$:

$$\begin{aligned}\lambda^* &= \operatorname{argmin}_{\lambda\in[0,1]}\|g - M\tilde{x}(\lambda)\| \\ &= \operatorname{argmin}_{\lambda\in[0,1]}\|g - M(\bar{x} + \lambda(\bar{p} - \bar{x}))\| = \operatorname{argmin}_{\lambda\in[0,1]}\|(1-\lambda)\bar{v} + \lambda\bar{w}\|.\end{aligned}$$

Notice that $\tilde{x}$ is a convex combination of the two admissible points $\bar{x}$ and $\bar{p}$ and therefore $\tilde{x}$ is also admissible. Also, λ^* above can be computed as the solution of the following simple constrained convex quadratic minimization problem:

$$\min_{\lambda\in[0,1]} \|(1-\lambda)\bar{v} + \lambda\bar{w}\|^2 = \min_{\lambda\in[0,1]} \lambda^2\|\bar{v} - \bar{w}\|^2 + 2\lambda(\bar{v}^t(\bar{w} - \bar{v})) + \|\bar{v}\|^2. \tag{20}$$

The closed-form solution of the program (20) is easily seen to be

$$\lambda^* = \min\left\{\frac{\bar{v}^t(\bar{v} - \bar{w})}{\|\bar{v} - \bar{w}\|^2}, 1\right\}. \tag{21}$$

The formal description of the algorithm is as follows:

Algorithm GVNA

- *Data:* (M, g, x^0) (where x^0 is an arbitrary admissible starting point).
- *Initialization:* The algorithm is initialized with x^0.
- *Iteration k, $k \geq 1$:* At the start of the iteration we have an admissible point x^{k-1} : $x^{k-1} \in C,\ \bar{u}^t x^{k-1} = 1$.

  Step 1 Compute $v^{k-1} = g - Mx^{k-1}$ (the residual).

  Step 2 Call the oracle to solve the following instance of (CSOP_C):

$$\begin{array}{ll} \min\limits_{p} (v^{k-1})^t(g - Mp) = \min\limits_{p} (v^{k-1})^t(g\bar{u}^t - M)p \\ \text{s.t.}\ \ p \in C & \text{s.t.}\ \ p \in C \\ \quad\ \bar{u}^t p = 1 & \quad\ \bar{u}^t p = 1. \end{array} \tag{22}$$

  Let p^{k-1} be an optimal solution of the optimization problem (22) and $w^{k-1} = g - Mp^{k-1}$. Evaluate $STOP(\cdot)$. If $STOP(\cdot) = 1$, stop, return appropriate output.

Step 3 Else, let

$$\lambda^{k-1} = \text{argmin}_{\lambda\in[0,1]}\{\|g - M(x^{k-1} + \lambda(p^{k-1} - x^{k-1}))\|\} \tag{23}$$

$$= \min\left\{\frac{(v^{k-1})^t(v^{k-1} - w^{k-1})}{\|v^{k-1} - w^{k-1}\|^2}, 1\right\}$$

and

$$x^k = x^{k-1} + \lambda^{k-1}(p^{k-1} - x^{k-1}).$$

Step 4 Let $k \leftarrow k + 1$, go to Step 1.

Note that the above description is rather generic; to apply the algorithm we have to specify the function $STOP(\cdot)$ to be used in Step 2. Some examples of function $STOP(\cdot)$ that will be used in this paper are:

1. $STOP1(v^{k-1}, w^{k-1}) = 1$ if $(v^{k-1})^t w^{k-1} > 0$, $STOP1 = 0$ otherwise. If the vectors v^{k-1}, w^{k-1} satisfy termination criterion $STOP1$, then it can be easily verified that the vector $s = -\frac{v^{k-1}}{\|v^{k-1}\|}$ is a solution to the alternative system (A) (see Proposition 5). Therefore, algorithm GVNA with $STOP = STOP1$ will terminate only if the system (P) is infeasible.
2. $STOP2(v^{k-1}, w^{k-1}) = 1$ if $(v^{k-1})^t w^{k-1} > \frac{\|v^{k-1}\|^2}{2}$, $STOP2 = 0$ otherwise. This termination criterion is a stronger version of the previous one.
3. $STOP3(v^{k-1}, w^{k-1}, k) = 1$ if $(v^{k-1})^t w^{k-1} > 0$ or $k \geq I$, where I is some pre-specified integer, $STOP3 = 0$ otherwise. This termination criterion is essentially equivalent to $STOP1$, but it ensures finite termination (in no more than I iterations) regardless of the status of (P).

Proposition 5. *Suppose v^{k-1} and w^{k-1} are as defined in Steps 1 and 2 of algorithm GVNA. If $(v^{k-1})^t w^{k-1} > 0$, then (A) has a solution and so (P) is infeasible.*

Proof. By definition of w^{k-1},

$$0 < (v^{k-1})^t w^{k-1} = (v^{k-1})^t(g\bar{u}^t - M)p^{k-1} \leq (v^{k-1})^t(g\bar{u}^t - M)p$$

for any $p \in C$, $\bar{u}^t p = 1$. Hence, $(g\bar{u}^t - M)^t v^{k-1} \in \text{int}\, C^*$ and $s = -\frac{v^{k-1}}{\|v^{k-1}\|}$ is a solution of (A).

□

Analogous to the von Neumann algorithm of [6] and [7], we regard algorithm GVNA as "elementary" in that the algorithm does not perform any sophisticated computations at each iteration (each iteration must perform a few matrix-vector and vector-vector multiplications and solve an instance of (CSOP_C)). Furthermore the work per iteration will be low so long as the number of operations performed by the oracle is small. Each iteration of algorithm GVNA requires at most

$$T_C + O(mn)$$

operations, where T_C is the number of operations performed by the oracle. The term $O(mn)$ derives from counting the matrix-vector and vector-vector multiplications. The number of operations required to perform these multiplications can be significantly reduced if M and g are sparse.

It can be easily seen that the size of the residual $\|v^k\|$ is non-increasing, since the interval of the line-search in (23) includes $\lambda = 0$. In fact, the size of the residual will decrease when either of the three termination criteria above is used. The rate of decrease depends on the termination criterion used and on the status of the system (P). In the rest of this section we present three lemmas that provide upper bounds on the size of the residual throughout the algorithm. The first result is a generalization of Dantzig's convergence result [6].

Lemma 1 (Dantzig [6]). *If algorithm GVNA with* $STOP = STOP1$ *(or* $STOP = STOP3$*) has performed k (complete) iterations, then*

$$\|v^k\| \leq \frac{\|M - g\bar{u}^t\|}{\beta_C \sqrt{k}}. \tag{24}$$

Proof. First note that if x is any admissible point (i.e., $x \in C$ and $\bar{u}^t x = 1$), then $\|x\| \leq \frac{\bar{u}^t x}{\beta_C} = \frac{1}{\beta_C}$, and so

$$\|g - Mx\| = \|(g\bar{u}^t - M)x\| \leq \|M - g\bar{u}^t\| \cdot \|x\| \leq \frac{\|M - g\bar{u}^t\|}{\beta_C}. \tag{25}$$

From the discussion preceding the formal statement of the algorithm, all iterates of the algorithm are admissible, so that $x^k \in C$ and $\bar{u}^t x^k = 1$ for all k. We prove the bound on the norm of the residual by induction on k.

For $k = 1$,

$$\|v^1\| = \|g - Mx^1\| \leq \frac{\|M - g\bar{u}^t\|}{\beta_C} = \frac{\|M - g\bar{u}^t\|}{\beta_C \sqrt{1}},$$

where the inequality above derives from (25).

Next suppose by induction that $\|v^{k-1}\| \leq \frac{\|M - g\bar{u}^t\|}{\beta_C \sqrt{k-1}}$. At the end of iteration k we have

$$\begin{aligned} \|v^k\| = \|g - Mx^k\| &= \|(1 - \lambda^{k-1})(g - Mx^{k-1}) + \lambda^{k-1}(g - Mp^{k-1})\| \\ &= \|(1 - \lambda^{k-1})v^{k-1} + \lambda^{k-1} w^{k-1}\|, \end{aligned} \tag{26}$$

where p^{k-1} and w^{k-1} were computed in Step 2. Recall that λ^{k-1} was defined in Step 3 as the minimizer of $\|(1 - \lambda)v^{k-1} + \lambda w^{k-1}\|$ over all $\lambda \in [0, 1]$. Therefore, in order to obtain an upper bound on $\|v^k\|$, we can substitute any $\lambda \in [0, 1]$ into (26). We will substitute $\lambda = \frac{1}{k}$. Making this substitution, we obtain:

$$\|v^k\| \leq \left\| \frac{k-1}{k} v^{k-1} + \frac{1}{k} w^{k-1} \right\| = \frac{1}{k} \|(k-1)v^{k-1} + w^{k-1}\|. \tag{27}$$

Squaring (27) yields:

$$\|v^k\|^2 \le \frac{1}{k^2}\left((k-1)^2\|v^{k-1}\|^2 + \|w^{k-1}\|^2 + 2(k-1)(v^{k-1})^t(w^{k-1})\right). \tag{28}$$

Since the algorithm did not terminate at Step 2, the termination criterion was not met, i.e., in the case $STOP = STOP1$ (or $STOP = STOP3$), $(v^{k-1})^t w^{k-1} \le 0$. Also, since p^{k-1} is admissible, $\|w^{k-1}\| = \|g - Mp^{k-1}\| \le \frac{\|M - g\bar{u}^t\|}{\beta_C}$. Combining these results with the inductive bound on $\|v^{k-1}\|$ and substituting into (28) above yields

$$\|v^k\|^2 \le \frac{1}{k^2}\left((k-1)^2\frac{\|M-g\bar{u}^t\|^2}{\beta_C^2(k-1)} + \frac{\|M-g\bar{u}^t\|^2}{\beta_C^2}\right) = \frac{1}{k}\cdot\frac{\|M-g\bar{u}^t\|^2}{\beta_C^2}.$$

□

We now develop another line of analysis of the algorithm, which will be used when the problem (P) is "well-posed." Let

$$\mathcal{H} = \mathcal{H}_M = \{Mx : x \in C,\ \bar{u}^t x = 1\}, \tag{29}$$

and notice that (P) is feasible precisely when $g \in \mathcal{H}$. Define

$$r = r(M, g) = \inf\{\|g - h\| : h \in \partial\mathcal{H}\}. \tag{30}$$

As it turns out, the quantity r plays a crucial role in analyzing the complexity of algorithm GVNA.

Observe that $r(M, g) = 0$ precisely when the vector g is on the boundary of the set $\mathcal{H}$. Thus, when $r = 0$, the problem (P) has a feasible solution, but arbitrarily small changes in the data (M, g) can yield instances of (P) that have no feasible solution. Therefore when $r = 0$ we can rightfully call the problem (P) unstable, or in the language of data perturbation and condition numbers, the problem (P) is "ill-posed." We will refer to the system (P) as being "well-posed" when $r > 0$.

Notice that both $\mathcal{H} = \mathcal{H}_M$ and $r = r(M, g)$ are specific to a given data instance (M, g) of (P), i.e., their definitions depend on the problem data M and g. We will, however, often omit problem data M and g from the notation for $\mathcal{H} = \mathcal{H}_M$ and $r = r(M, g)$. It should be clear from the context which data instance we are referring to.

In light of Remark 3, when (P) has a feasible solution, $r(M, g)$ can be interpreted as the radius of the largest ball centered at g and contained in the set $\mathcal{H}$.

We now present an analysis of the performance of algorithm GVNA in terms of the quantity $r = r(M, g)$.

Proposition 6. *Suppose that (P) has a feasible solution. Let v^k be the residual at point x^k, and let p^k be the direction found in Step 2 of the algorithm at iteration $k+1$. Then $(v^k)^t(g - Mp^k) + r(M, g)\|v^k\| \le 0$.*

Proof. If $v^k = 0$, the result follows trivially. Suppose $v^k \ne 0$. By definition of $r(M, g)$, there exists a point $h \in \mathcal{H}$ such that $g - h + r(M, g)\frac{v^k}{\|v^k\|} = 0$. By the definition of $\mathcal{H}$, $h = Mx$ for some admissible point x. It follows that

$$g - Mx = -r(M, g)\frac{v^k}{\|v^k\|}.$$

Recall that $p^k \in \operatorname{argmin}_p\{(v^k)^t(g - Mp) : p \in C,\ \bar{u}^t p = 1\}$. Therefore,

$$(v^k)^t(g - Mp^k) \leq (v^k)^t(g - Mx) = -(v^k)^t r(M, g)\frac{v^k}{\|v^k\|} = -r(M, g)\|v^k\|,$$

and rearranging yields

$$(v^k)^t(g - Mp^k) + r(M, g)\|v^k\| \leq 0.$$

□

Proposition 6 is used to prove the following linear convergence rate for algorithm GVNA:

Lemma 2. *Suppose the system (P) is feasible, and that $r(M, g) > 0$. If GVNA with $STOP = STOP1$ (or $STOP = STOP3$) has performed k (complete) iterations, then*

$$\|v^k\| \leq \|v^0\| e^{-\frac{k}{2}\left(\frac{\beta_C r(M,g)}{\|M - g\bar{u}^t\|}\right)^2}. \tag{31}$$

Proof. Let $\bar{x}$ be the current iterate of GVNA. Furthermore, let $\bar{v} = g - M\bar{x}$ be the residual at the point $\bar{x}$, $\bar{p}$ be the solution of the problem (CSOP$_C$), and $\bar{w} = g - M\bar{p}$. Suppose that the algorithm has not terminated at the current iteration, and $\tilde{x} = \bar{x} + \lambda^*(\bar{p} - \bar{x})$ is the next iterate and $\tilde{v}$ is the residual at $\tilde{x}$. Then

$$\|\tilde{v}\|^2 = \|(1 - \lambda^*)\bar{v} + \lambda^*\bar{w}\|^2 = (\lambda^*)^2\|\bar{v} - \bar{w}\|^2 + 2\lambda^*\bar{v}^t(\bar{w} - \bar{v}) + \|\bar{v}\|^2, \tag{32}$$

where $\lambda^* = \min\left\{\frac{\bar{v}^t(\bar{v} - \bar{w})}{\|\bar{v} - \bar{w}\|^2}, 1\right\}$. Since the algorithm has not terminated at Step 2, the termination criterion has not been satisfied, i.e., in the case of $STOP = STOP1$ (or $STOP = STOP3$), $\bar{v}^t\bar{w} \leq 0$. Therefore

$$\bar{v}^t(\bar{v} - \bar{w}) \leq \|\bar{v}\|^2 - \bar{v}^t\bar{w} + (\|\bar{w}\|^2 - \bar{v}^t\bar{w}) = \|\bar{v} - \bar{w}\|^2,$$

so that $\frac{\bar{v}^t(\bar{v} - \bar{w})}{\|\bar{v} - \bar{w}\|^2} \leq 1$ and $\lambda^* = \frac{\bar{v}^t(\bar{v} - \bar{w})}{\|\bar{v} - \bar{w}\|^2}$. Substituting this value of λ^* into (32) yields:

$$\|\tilde{v}\|^2 = \frac{\|\bar{v}\|^2\|\bar{w}\|^2 - (\bar{v}^t\bar{w})^2}{\|\bar{v} - \bar{w}\|^2}. \tag{33}$$

Recall from Proposition 6 that $\bar{v}^t\bar{w} \leq -r(M, g)\|\bar{v}\|$. Thus, $\|\bar{v}\|^2(\|\bar{w}\|^2 - r(M, g)^2)$ is an upper bound on the numerator of (33). Also, $\|\bar{v} - \bar{w}\|^2 = \|\bar{v}\|^2 + \|\bar{w}\|^2 - 2\bar{v}^t\bar{w} \geq \|\bar{w}\|^2$. Substituting this into (33) yields

$$\begin{aligned}\|\tilde{v}\|^2 \leq \frac{\|\bar{v}\|^2(\|\bar{w}\|^2 - r(M, g)^2)}{\|\bar{w}\|^2} &= \left(1 - \frac{r(M, g)^2}{\|\bar{w}\|^2}\right)\|\bar{v}\|^2 \\ &\leq \left(1 - \left(\frac{\beta_C r(M, g)}{\|g\bar{u}^t - M\|}\right)^2\right)\|\bar{v}\|^2,\end{aligned}$$

where the last inequality derives from (25). Applying the inequality $1 - t \leq e^{-t}$ for $t = \left(\frac{\beta_C r(M,g)}{\|g\bar{u}^t - M\|}\right)^2$, we obtain:

$$\|\tilde{v}\|^2 \leq \|\bar{v}\|^2 e^{-\left(\frac{\beta_C r(M,g)}{\|g\bar{u}^t - M\|}\right)^2},$$

or, substituting $\bar{v} = v^{k-1}$ and $\tilde{v} = v^k$,

$$\|v^k\| \leq \|v^{k-1}\| e^{-\frac{1}{2}\left(\frac{\beta_C r(M,g)}{\|g\bar{u}^t - M\|}\right)^2}. \tag{34}$$

Applying (34) inductively, we can bound the size of the residual $\|v^k\|$ by

$$\|v^k\| \leq \|v^0\| e^{-\frac{k}{2}\left(\frac{\beta_C r(M,g)}{\|g\bar{u}^t - M\|}\right)^2}.$$

□

We now establish a bound on the size of the residual for $STOP = STOP2$.

Lemma 3. *If GVNA with* $STOP = STOP2$ *has performed k (complete) iterations, then*

$$\|v^k\| \leq \frac{4\|M - g\bar{u}^t\|}{\beta_C \sqrt{k}}.$$

Proof. Let $\bar{x}$ be the current iterate of GVNA. Furthermore, let $\bar{v} = g - M\bar{x}$ be the residual at the point $\bar{x}$, $\bar{p}$ be the solution of the problem (CSOP$_C$) and $\bar{w} = g - M\bar{p}$. Suppose that the algorithm has not terminated at the current iteration, and $\tilde{x} = \bar{x} + \lambda^*(\bar{p} - \bar{x})$ is the next iterate and $\tilde{v}$ is the residual at $\tilde{x}$. Then

$$\|\tilde{v}\|^2 = \|(1 - \lambda^*)\bar{v} + \lambda^*\bar{w}\|^2 = (\lambda^*)^2\|\bar{v} - \bar{w}\|^2 + 2\lambda^*\bar{v}^t(\bar{w} - \bar{v}) + \|\bar{v}\|^2, \tag{35}$$

where λ^* is given by (21). Consider two cases:

Case 1: $\|\bar{w}\|^2 \leq \bar{w}^t\bar{v}$. It can be easily shown that in this case $\lambda^* = 1$. Substituting this value of λ^* into (35), algebraic manipulations yield

$$\|\tilde{v}\|^2 = \|\bar{w}\|^2 \leq \bar{w}^t\bar{v} \leq \frac{\|\bar{v}\|^2}{2} = \|\bar{v}\|^2 - \frac{\|\bar{v}\|^2}{2} \leq \|\bar{v}\|^2 - \frac{\|\bar{v}\|^4\beta_C^2}{16\|M - g\bar{u}^t\|^2}. \tag{36}$$

The second inequality in (36) follows from the assumption that the algorithm did not terminate at the present iteration, i.e., in the case of $STOP = STOP2$, $\bar{v}^t\bar{w} \leq \frac{\|\bar{v}\|^2}{2}$. The last inequality follows since

$$\|\bar{v}\|^2 \leq \frac{\|M - g\bar{u}^t\|^2}{\beta_C^2} \leq \frac{8\|M - g\bar{u}^t\|^2}{\beta_C^2}.$$

The need for the last inequality may not be immediately clear at this stage, but will become more apparent later in this proof.

Case 2: $\|\bar{w}\|^2 \geq \bar{w}^t\bar{v}$. It can be easily shown that in this case $\lambda^* = \frac{\bar{v}^t(\bar{v}-\bar{w})}{\|\bar{v}-\bar{w}\|^2}$. Substituting this value of λ^* into (35) yields:

$$\|\tilde{v}\|^2 = \|\bar{v}\|^2 - \frac{(\bar{v}^t(\bar{w}-\bar{v}))^2}{\|\bar{w}-\bar{v}\|^2}.$$

Since $\bar{v}^t\bar{w} \leq \frac{\|\bar{v}\|^2}{2}$, we have:

$$\bar{v}^t(\bar{v}-\bar{w}) \geq \frac{\|\bar{v}\|^2}{2},$$

so that

$$\|\tilde{v}\|^2 \leq \|\bar{v}\|^2 - \frac{\|\bar{v}\|^4}{4\|\bar{w}-\bar{v}\|^2} \leq \|\bar{v}\|^2 - \frac{\|\bar{v}\|^4\beta_C^2}{16\|M-g\bar{u}^t\|^2},$$

since

$$\|\bar{w}-\bar{v}\|^2 \leq \|\bar{v}\|^2 + \|\bar{w}\|^2 + 2\|\bar{v}\|\cdot\|\bar{w}\| \leq \frac{4\|M-g\bar{u}^t\|^2}{\beta_C^2}$$

(the last inequality results from an application of (25) for $\|\bar{v}\| = \|g - M\bar{x}\|$ and $\|\bar{w}\| = \|g - M\bar{p}\|$).

Combining Case 1 and Case 2, we conclude that

$$\|\tilde{v}\|^2 \leq \|\bar{v}\|^2 - \frac{\|\bar{v}\|^4}{\gamma^2}, \text{ where } \gamma \stackrel{\triangle}{=} \frac{4\|M-g\bar{u}^t\|}{\beta_C}. \tag{37}$$

Next, we establish (using induction) the following relation, from which the statement of the lemma will follow: if the algorithm has performed k (complete) iterations, then

$$\|v^k\|^2 \leq \frac{\gamma^2}{k}. \tag{38}$$

First, note that $\|v^1\|^2 \leq \frac{\|M-g\bar{u}^t\|^2}{\beta_C^2} \leq \frac{\gamma^2}{1}$, thus establishing (38) for $k = 1$. Suppose that (38) holds for $k \geq 1$. Then, using the relationship for $\tilde{v}$ and $\bar{v}$ established above with $\tilde{v} = v^{k+1}$ and $\bar{v} = v^k$, we have:

$$\|v^{k+1}\|^2 \leq \|v^k\|^2 - \frac{\|v^k\|^4}{\gamma^2},$$

or, dividing by $\|v^{k+1}\|^2\cdot\|v^k\|^2$,

$$\frac{1}{\|v^k\|^2} \leq \frac{1}{\|v^{k+1}\|^2} - \frac{\|v^k\|^2}{\|v^{k+1}\|^2\gamma^2} \leq \frac{1}{\|v^{k+1}\|^2} - \frac{1}{\gamma^2}.$$

Therefore,

$$\frac{1}{\|v^{k+1}\|^2} \geq \frac{1}{\|v^k\|^2} + \frac{1}{\gamma^2} \geq \frac{k}{\gamma^2} + \frac{1}{\gamma^2},$$

and so

$$\|v^{k+1}\|^2 \leq \frac{\gamma^2}{k+1},$$

thus establishing the relation (38), which completes the proof of the lemma.

□

4. Elementary algorithms for homogeneous conic linear systems

In this section we develop and analyze two elementary algorithms for homogeneous conic linear systems: algorithm HCI (for Homogeneous Conic Inequalities) which solves systems of the form

$$\text{(HCI)} \quad M^t s \in \operatorname{int} C^*, \tag{39}$$

and algorithm HCE (for Homogeneous Conic Equalities) which solves systems of the form

$$\begin{array}{ll} \text{(HCE)} & Mw = 0, \\ & w \in C. \end{array} \tag{40}$$

Here the notation is the same as in Sect. 3, and we make the following assumption:

Assumption 2. *$C \subset X$ is a regular cone. The width τ_C of the cone C and the coefficient of linearity β_C for the cone C, together with vectors $\bar{u}$ and u of Remark 2 and Proposition 3 are known and given. For $y \in Y$, $\|y\| = \|y\|_2$.*

Both algorithms HCI and HCE consist of calls to algorithm GVNA applied to transformations of the appropriate homogeneous system. Algorithms HCI and HCE will be used in Sect. 5 in the development of algorithm CLS for general conic linear system (FP$_d$).

4.1. Algorithm HCI for homogeneous conic inequality system (HCI)

In this subsection we will assume that the system (HCI) of (39) is feasible. We denote the set of solutions of (HCI) by S_M:

$$S_M \stackrel{\triangle}{=} \{s : M^t s \in \operatorname{int} C^*\}.$$

The solution s returned by algorithm HCI is "sufficiently interior" in the sense that the ratio $\frac{\|s\|_*}{\operatorname{dist}(s, \partial S_M)}$ is not excessively large. (The notion of sufficiently interior solutions is very similar to the notion of reliable solutions. However, we wish to reserve the appellation "reliable" for solutions and certificates of infeasibility of the system (FP$_d$).)

Observe that the system (HCI) of (39) is of the form (18) (with $g = 0$). (HCI) is the "alternative" system for the following problem:

$$\begin{array}{ll} \text{(PHCI)} & Mx = 0 \\ & x \in C \\ & \bar{u}^t x = 1, \end{array} \tag{41}$$

which is a system of the form (17). Following (30) we define

$$r(M, 0) \stackrel{\triangle}{=} \inf\{\|h\| : h \in \partial\mathcal{H}\}, \tag{42}$$

where, as in (29), $\mathcal{H} \stackrel{\triangle}{=} \{Mx : x \in C,\ \bar{u}^t x = 1\}$. Applying a separating hyperplane argument, we easily have the following result:

Proposition 7. *Suppose (HCI) of (39) is feasible. Then (PHCI) of (41) is infeasible and* $r(M,0) = \min\{\|Mx\| : x \in C,\ \bar{u}^t x = 1\} > 0$.

Algorithm HCI, described and analyzed below, consists of a single application of algorithm GVNA to the system (PHCI) and returns as output a sufficiently interior solution of the system (HCI).

Algorithm HCI

- *Data: M*
- Run algorithm GVNA with $STOP = STOP2$ on the data set $(M, 0, x^0)$ (where x^0 is an arbitrary admissible starting point). Let $\bar{v}$ be the residual at the last iteration of algorithm GVNA.
- Define $s \stackrel{\triangle}{=} -\frac{\bar{v}}{\|\bar{v}\|}$. Return s.

Theorem 1. *Suppose (HCI) is feasible. Algorithm HCI will terminate in at most*

$$\left\lfloor \frac{16\|M\|^2}{\beta_C^2 r(M,0)^2} \right\rfloor \tag{43}$$

iterations of algorithm GVNA.

Let s be the output of algorithm HCI. Then $s \in S_M$ *and*

$$\frac{\|s\|_*}{\operatorname{dist}(s, \partial S_M)} \leq \frac{2\|M\|}{\beta_C r(M,0)}. \tag{44}$$

Proof. Suppose that algorithm GVNA (called in algorithm HCI) has completed k iterations. From Lemma 3 we conclude that

$$\|v^k\| \leq \frac{4\|M\|}{\beta_C \sqrt{k}},$$

where $v^k = -Mx^k$ is the residual after k iterations. From Proposition 7, $r(M,0) \leq \|Mx\|$ for any admissible point x. Therefore,

$$r(M,0) \leq \|v^k\| \leq \frac{4\|M\|}{\beta_C \sqrt{k}}.$$

Rearranging yields

$$k \leq \frac{16\|M\|^2}{\beta_C^2 r(M,0)^2},$$

from which the first part of the theorem follows.

Next, observe that $\|s\|_* = 1$. Therefore, to establish the second part of the theorem, we need to show that $\operatorname{dist}(s, \partial S_M) \geq \frac{\beta_C r(M,0)}{2\|M\|}$. Equivalently, we need to show that for any $q \in Y^*$ such that $\|q\|_* \leq 1$, $M^t\left(s + \frac{\beta_C r(M,0)}{2\|M\|} q\right) \in C^*$. Let p be an arbitrary vector satisfying $p \in C$, $\bar{u}^t p = 1$. Then

$$\left(M^t\left(s + \frac{\beta_C r(M,0)}{2\|M\|} q\right)\right)^t p = s^t M p + \frac{\beta_C r(M,0)}{2\|M\|} q^t M p. \tag{45}$$

Observe that by definition of s

$$s^t Mp = \frac{-\bar{v}^t Mp}{\|\bar{v}\|} \geq \frac{\bar{v}^t w^{k-1}}{\|\bar{v}\|} > \frac{\|\bar{v}\|}{2},$$

where $\bar{v} = v^{k-1}$ is the residual at the last iteration of algorithm GVNA. (The first inequality follows since p is an admissible point, and the second inequality follows from the fact that the termination criterion of $STOP2$ is satisfied at the last iteration.) On the other hand,

$$\frac{\beta_C r(M,0)}{2\|M\|} q^t Mp \geq -\frac{\beta_C r(M,0)}{2\|M\|} \|q\|_* \cdot \|M\| \cdot \|p\| \geq -\frac{r(M,0)}{2}.$$

Substituting the above two bounds into (45), we conclude that

$$\left(M^t \left(s + \frac{\beta_C r(M,0)}{2\|M\|} q \right) \right)^t p > \frac{\|\bar{v}\|}{2} - \frac{r(M,0)}{2} \geq 0.$$

□

4.2. Algorithm HCE for homogeneous conic equality system (HCE)

We denote the set of solutions of (HCE) of (40) by W_M, i.e.,

$$W_M \stackrel{\triangle}{=} \{w : Mw = 0,\ w \in C\}.$$

We assume in this subsection that (HCE) is feasible and M has full rank. The solution w returned by algorithm HCE is "sufficiently interior" in the sense that the ratio $\frac{\|w\|}{\text{dist}(w, \partial C)}$ is not excessively large. (The system (HCE) has a trivial solution $w = 0$. However this solution is not a sufficiently interior solution, since it is contained in the boundary of the cone C).

We define $\tilde{\mathcal{H}} = \{Mx : x \in C,\ \|x\| \leq 1\}$ (note the similarity with $\mathcal{H}_M$ of (29)), and

$$\rho(M) \stackrel{\triangle}{=} \text{dist}(0, \partial\tilde{\mathcal{H}}) = \max\{r : B(0,r) \subset \tilde{\mathcal{H}}\}. \tag{46}$$

The following remark summarizes some important facts about $\rho(M)$:

Remark 4. Suppose $\rho(M) > 0$. Then the set $\{w \in W_M : w \neq 0\}$ is non-empty, and M has full rank. Moreover, $\rho(M) \leq \|M\|$ and

$$\|(MM^t)^{-1}\| \leq \frac{1}{\rho(M)^2}. \tag{47}$$

This follows from the observation that $\rho(M)^2 \leq \lambda_1(MM^t)$, where $\lambda_1(MM^t)$ denotes the smallest eigenvalue of the matrix MM^t.

We will assume for the rest of this subsection that $\rho(M) > 0$. Then the second statement of Remark 4 implies that the earlier assumption that M has full rank is satisfied. In order to obtain a sufficiently interior solution of (HCE) we will construct a transformation of the system (HCE) which has the form (17), and its solutions can be transformed into sufficiently interior solutions of the system (HCE). The next subsection contains the analysis of the transformation, and its results are used to develop algorithm HCE in the following subsection.

4.2.1. Properties of a parameterized conic system of equalities in compact form. In this subsection we work with a compact-form system

$$\begin{aligned} (\mathrm{HCE}_0)\quad Mx &= 0 \\ x &\in C \\ \bar{u}^t x &= 1. \end{aligned} \tag{48}$$

The system (HCE_0) is derived from the system (HCE) by adding a compactifying constraint $\bar{u}^t x = 1$. Remark 4 implies that when $\rho(M) > 0$ the system (HCE_0) is feasible.

We will consider systems arising from parametric perturbations of the right-hand side of (HCE_0). In particular, for a fixed vector $z \in Y$, we consider the perturbed compact-form system

$$\begin{aligned} (\mathrm{HCE}_\delta)\quad Mx &= \delta z \\ x &\in C \\ \bar{u}^t x &= 1, \end{aligned} \tag{49}$$

where the scalar $\delta \geq 0$ is the perturbation parameter (observe that (HCE_0) can be viewed as an instance of (HCE_δ) with the parameter $\delta = 0$, justifying the notation). Since the case when $z = 0$ is trivial (i.e., (HCE_δ) is equivalent to (HCE_0) for all values of δ), we assume that $z \neq 0$. The following lemma establishes an estimate on the range of values of δ for which the resulting system is feasible, and establishes bounds on the parameters of the system (HCE_δ) in terms of δ.

Before stating the lemma, we will restate some facts about the geometric interpretation of (HCE_δ) and the parameter $r(M, \delta z)$ of (30). Recall that the system (HCE_δ) is feasible precisely when $\delta z \in \mathcal{H} \stackrel{\triangle}{=} \{Mx : x \in C,\ \bar{u}^t x = 1\}$. Also, if the system ($\mathrm{HCE}_\delta$) is feasible, $r(M, \delta z)$ can be interpreted as the radius of the largest ball centered at δz and contained in $\mathcal{H}$. Moreover, using the inequality $\beta_C \|x\| \leq \bar{u}^t x \leq \|x\|$ for all $x \in C$, it follows that

$$\beta_C r(M, 0) \leq \rho(M) \leq r(M, 0).$$

Lemma 4. *Suppose (HCE_0) of (48) is feasible, and $z \in Y$, $z \neq 0$. Define*

$$\bar{\delta} = \max\{\delta : \textit{(HCE}_\delta\textit{) is feasible}\}. \tag{50}$$

Then $\frac{\rho(M)}{\|z\|} \leq \frac{r(M,0)}{\|z\|} \leq \bar{\delta} < +\infty$. Moreover, if $\rho(M) > 0$, then $\bar{\delta} > 0$, and for any $\delta \in [0, \bar{\delta}]$, the system ($\mathrm{HCE}_\delta$) is feasible and $\|M - \delta z \bar{u}^t\| \leq \|M\| + \delta\|z\|$ and $r(M, \delta z) \geq \left(\frac{\bar{\delta}-\delta}{\bar{\delta}}\right) \rho(M)$.

Proof. Since $\mathcal{H}$ is compact and z is nonzero, $\bar{\delta}$ is well defined and finite. Note that the definition of $\bar{\delta}$ implies that $\bar{\delta} z \in \partial\mathcal{H}$. To establish the lower bound on $\bar{\delta}$, note that for any $y \in Y$ such that $\|y\| \leq 1$, $r(M, 0)y \in \mathcal{H}$. Therefore, if we take $y = \frac{z}{\|z\|}$, we have $\frac{r(M,0)}{\|z\|} z \in \mathcal{H}$, and so ($\mathrm{HCE}_\delta$) is feasible for $\delta = \frac{r(M,0)}{\|z\|}$. Hence, $\bar{\delta} \geq \frac{r(M,0)}{\|z\|} \geq \frac{\rho(M)}{\|z\|}$.

The bound on $\|M - \delta z \bar{u}^t\|$ is a simple application of the triangle inequality for the operator norm, i.e., $\|M - \delta z \bar{u}^t\| \leq \|M\| + \delta\|z\| \cdot \|\bar{u}\|_* = \|M\| + \delta\|z\|$.

Finally, suppose that $\rho(M) > 0$. Then $\bar{\delta} > 0$. Let $\delta \in [0, \bar{\delta}]$ be some value of the perturbation parameter. Since $\delta \leq \bar{\delta}$, the system (HCE$_\delta$) is feasible. To establish the lower bound on $r(M, \delta z)$ stated in the lemma, it is sufficient to show that a ball of radius $\frac{\bar{\delta}-\delta}{\bar{\delta}} r(M, 0)$ centered at δz is contained in $\mathcal{H}$. Suppose $y \in Y$ is such that $\|y\| \leq 1$. As noted above, $\bar{\delta} z \in \mathcal{H}$ and $r(M, 0)y \in \mathcal{H}$. Therefore,

$$\delta z + \frac{\bar{\delta} - \delta}{\bar{\delta}} r(M, 0)y = \frac{\delta}{\bar{\delta}}(\bar{\delta} z) + \left(1 - \frac{\delta}{\bar{\delta}}\right)(r(M, 0)y) \in \mathcal{H},$$

since the above is a convex combination of $\bar{\delta} z$ and $r(M, 0)y$. Therefore, $r(M, \delta z) \geq \frac{\bar{\delta}-\delta}{\bar{\delta}} r(M, 0) \geq \frac{\bar{\delta}-\delta}{\bar{\delta}} \rho(M)$, which concludes the proof.

□

We now consider the system (HCE$_\delta$) of (49) with the vector $z \stackrel{\triangle}{=} -Mu$, where u is as specified in Assumption 2. The system (HCE$_\delta$) becomes

$$\begin{array}{ll} \text{(HCE}_\delta\text{)} & Mx = -\delta Mu \\ & x \in C \\ & \bar{u}^t x = 1. \end{array} \tag{51}$$

The following proposition indicates how approximate solutions of the system (HCE$_\delta$) of (51) can be used to obtain sufficiently interior solutions of the system (HCE).

Proposition 8. *Suppose $\rho(M) > 0$ and $\delta > 0$. Suppose further that x is an admissible point for (HCE$_\delta$), and in addition x satisfies*

$$\|Mx + \delta Mu\| \leq \frac{1}{2}\delta\tau_C \frac{\rho(M)^2}{\|M\|}.$$

Define

$$w \stackrel{\triangle}{=} (I - M^t(MM^t)^{-1}M)(x + \delta u). \tag{52}$$

Then $Mw = 0$ and

$$\|w - (x + \delta u)\| \leq \frac{1}{2}\delta\tau_C \tag{53}$$

which implies that $w \in C$, $\text{dist}(w, \partial C) \geq \frac{1}{2}\delta\tau_C$, and $\|w\| \leq \frac{1}{2}\delta\tau_C + \frac{1}{\beta_C} + \delta$.

Proof. First, observe that w satisfies $Mw = 0$ by definition (52). To demonstrate (53) we apply the definition (52) of w to obtain

$$\|w - (x + \delta u)\| = \|M^t(MM^t)^{-1}M(x + \delta u)\| \leq \|M\| \cdot \|(MM^t)^{-1}\| \cdot \|M(x + \delta u)\|$$

$$\leq \frac{\delta\tau_C\rho(M)^2 \cdot \|M\| \cdot \|(MM^t)^{-1}\|}{2\|M\|} = \frac{\delta\tau_C\rho(M)^2 \cdot \|(MM^t)^{-1}\|}{2} \leq \frac{\delta\tau_C}{2},$$

since $\|(MM^t)^{-1}\| \leq \frac{1}{\rho(M)^2}$ from Remark 4.

The last three statements of the proposition are direct consequences of (53). Notice that $B(x+\delta u, \delta\tau_C) \subset C$ since $B(u, \tau_C) \subset C$ and $x \in C$. Combining this with (53) and the triangle inequality for the norm we conclude that $w \in C$ and $\text{dist}(w, \partial C) \geq \frac{1}{2}\delta\tau_C$. Also,

$$\|w\| \leq \|w - (x+\delta u)\| + \|x+\delta u\| \leq \frac{1}{2}\delta\tau_C + \frac{1}{\beta_C} + \delta,$$

which completes the proof.

□

Notice that w defined by (52) is the projection of $x+\delta u$ onto the set $\{w : Mw = 0\}$ with respect to the Euclidean norm on the space X. Although the norm on the space X may be different from the Euclidean norm, we will refer to the point w defined by (52) as the projection of $x+\delta u$. It is interesting to note that it is not necessary to have $\delta \leq \bar{\delta}$ for Proposition 8 to be applicable.

4.2.2. Algorithm HCE. Algorithm HCE applies algorithm GVNA to a sequence of problem (HCE$_\delta$) of (51) with decreasing values of δ, until the output provides a sufficiently interior solution of (HCE).

The formal statement of algorithm HCE is as follows:

Algorithm HCE

– *Data: M*
– *Iteration k*, $k \geq 1$

Step 1 $\delta = \delta^k \stackrel{\triangle}{=} 2^{1-k}$, compute $I(\delta)$:

$$I(\delta) \stackrel{\triangle}{=} \left\lceil \frac{9}{2\beta_C^2\delta^2} \ln\left(\frac{1}{2\tau_C\delta^2}\left(1 + \frac{1}{\beta_C\delta}\right)\right)\right\rceil. \tag{54}$$

Step 2 Run GVNA with $STOP = STOP3$ with $I = I(\delta)$ on the data set $(M, -\delta Mu, x^0)$ (where x^0 is an arbitrary admissible starting point).

Step 3 Let x be the last iterate of GVNA in Step 2.
Set $w = (I - M^t(MM^t)^{-1}M)(x+\delta u)$. If $\|w - (x+\delta u)\| \leq \frac{1}{2}\tau_C\delta$, stop.
Return w.
Else, set $k \leftarrow k+1$ and repeat Step 1.

The following proposition states that when $\rho(M) > 0$ algorithm HCE will terminate and return as output a sufficiently interior solution of (HCE).

Theorem 2. *Suppose (HCE) satisfies $\rho(M) > 0$. Algorithm HCE will terminate in at most*

$$\left\lceil \log_2\left(\frac{\|M\|}{\rho(M)}\right)\right\rceil + 2 \tag{55}$$

iterations, performing at most

$$\frac{4}{3}\left\lceil \frac{216\|M\|^2}{\rho(M)^2\beta_C^2} \ln\left(\frac{40\|M\|}{\rho(M)\tau_C\beta_C}\right)\right\rceil + \left\lceil \log_2\left(\frac{\|M\|}{\rho(M)}\right)\right\rceil + 2 \tag{56}$$

iterations of algorithm GVNA.

Algorithm HCE will return a vector $w \in X$ with the following properties:

1. $w \in W_M$,
2. $\text{dist}(w, \partial C) \geq \frac{\tau_C \rho(M)}{8\|M\|}$,
3. $\|w\| \leq \frac{5}{2\beta_C}$,
4. $\frac{\|w\|}{\text{dist}(w,\partial C)} \leq \frac{11\|M\|}{\rho(M)\beta_C\tau_C}$.

Proof. We begin by establishing the maximum number of iterations algorithm HCE will perform. Suppose that x is an admissible point for the system (HCE$_\delta$) for some value $\delta > 0$. The residual at point x is defined in algorithm GVNA as $v = -\delta Mu - Mx = -M(x + \delta u)$. From Proposition 8, having a residual with a small norm will guarantee that the projection w of the point $x+\delta u$ will satisfy the property $\|w-(x+\delta u)\| \leq \frac{1}{2}\tau_C\delta$. In particular, it is sufficient to have $\|v\| \leq \epsilon$ with

$$\epsilon = \frac{1}{2}\delta\tau_C\frac{\rho(M)^2}{\|M\|}. \tag{57}$$

We now argue that if $\delta \leq \frac{1}{2}\frac{\rho(M)}{\|M\|}$, then Step 2 of algorithm HCE will terminate in $I(\delta)$ iterations and produce an iterate with the size of the residual no larger than ϵ given by (57).

Suppose $0 < \delta \leq \frac{1}{2}\frac{\rho(M)}{\|M\|}$. Let $\bar{\delta}$ be as defined in (50). Applying Lemma 4 for $z = -Mu$ we conclude that the system (HCE$_\delta$) is feasible for any $\delta \in [0, \bar{\delta}]$, and $\bar{\delta} \geq \frac{\rho(M)}{\|Mu\|} \geq \frac{\rho(M)}{\|M\|} \geq 2\delta$. Hence the system (HCE$_\delta$) is feasible, and furthermore

$$\|M + \delta Mu\bar{u}^t\| \leq (1+\delta)\|M\| \leq \frac{3}{2}\|M\|$$

(since $\delta \leq \frac{1}{2}$), and

$$r(M, -\delta Mu) \geq \left(\frac{\bar{\delta}-\delta}{\bar{\delta}}\right)\rho(M) \geq \frac{1}{2}\rho(M).$$

Since the system (HCE$_\delta$) is feasible, from Proposition 5 it must be true that algorithm GVNA with $STOP = STOP3$ will perform $I = I(\delta)$ iterations, where

$$I(\delta) \stackrel{\Delta}{=} \left\lceil \frac{9}{2\beta_C^2\delta^2}\ln\left(\frac{1}{2\tau_C\delta^2}\left(1+\frac{1}{\beta_C\delta}\right)\right)\right\rceil \geq \frac{18\|M\|^2}{\rho(M)^2\beta_C^2}\ln\left(\frac{2\|M\|^2}{\rho(M)^2\tau_C}\left(1+\frac{1}{\beta_C\delta}\right)\right), \tag{58}$$

since $\delta \leq \frac{1}{2}\frac{\rho(M)}{\|M\|}$. Applying Lemma 2 we conclude that after $I(\delta)$ iterations of GVNA the residual $v^{I(\delta)}$ satisfies:

$$\|v^{I(\delta)}\| \leq \|v^0\|e^{-\frac{I(\delta)}{2}\left(\frac{\beta_C r(M,-\delta Mu)}{\|M+\delta Mu\bar{u}^t\|}\right)^2} \leq \|Mx^0 + \delta Mu\|e^{-\frac{I(\delta)}{2}\left(\frac{\beta_C\rho(M)}{3\|M\|}\right)^2}$$

$$\leq \left(\frac{1}{\beta_C}+\delta\right)\|M\|e^{-\frac{9\|M\|^2}{\rho(M)^2\beta_C^2}\ln\left(\frac{2\|M\|^2}{\rho(M)^2\tau_C}\left(1+\frac{1}{\beta_C\delta}\right)\right)\cdot\left(\frac{\beta_C\rho(M)}{3\|M\|}\right)^2} = \frac{\rho(M)^2\tau_C\delta}{2\|M\|} = \epsilon.$$

We conclude that if $0 < \delta \leq \frac{1}{2}\frac{\rho(M)}{\|M\|}$, then algorithm GVNA of Step 2 of HCE will perform $I(\delta)$ iterations and w defined in Step 3 will satisfy the termination criterion of HCE.

In principle, algorithm HCE might terminate with a solution after as little as one iteration, if the point w defined in Step 3 of that iteration happens to be sufficiently close to the point $x + \delta u$. However, in the worst case algorithm HCE will continue iterating until the value of δ becomes small enough to guarantee (by the analysis above) that the corresponding iteration will produce a point satisfying the termination criterion. To make this argument more precise, recall that during the kth iteration of the algorithm HCE, $\delta = \delta^k = 2^{1-k}$. Hence, HCE is guaranteed to stop at (or before) the iteration during which value of δ falls below $\frac{1}{2}\frac{\rho(M)}{\|M\|}$ for the first time. In other words, the number of iterations of HCE that are performed is bounded above by

$$\min\left\{k : 2^{1-k} \leq \frac{1}{2}\frac{\rho(M)}{\|M\|}\right\}.$$

Therefore algorithm HCE will terminate in no more than

$$K = \left\lceil \log_2\left(\frac{\|M\|}{\rho(M)}\right)\right\rceil + 2 \tag{59}$$

iterations, which proves the first claim of the theorem. Also, notice that throughout the algorithm,

$$\delta^k > \frac{1}{4}\frac{\rho(M)}{\|M\|}. \tag{60}$$

To bound the total number of iterations of GVNA performed by HCE, we need to bound the sum of the corresponding $I(\delta)$'s:

$$\sum_{k=1}^{K} I(\delta^k) = \sum_{k=1}^{K}\left\lceil \frac{9\cdot 4^k}{8\beta_C^2}\ln\left(\frac{4^k}{8\tau_C}\left(1+\frac{2^{k-1}}{\beta_C}\right)\right)\right\rceil. \tag{61}$$

It can be shown by analyzing the geometric series $\sum_{k=1}^{K} 4^k$ that the sum in (61) satisfies $\sum_{k=1}^{K} I(\delta^k) \leq \frac{4}{3}I(\delta^K) + K$. Therefore

$$\begin{aligned}
\sum_{k=1}^{K} I(\delta^k) &\leq \frac{4}{3}\left\lceil \frac{9}{2\beta_C^2(\delta^K)^2}\ln\left(\frac{1}{2\tau_C(\delta^K)^2}\left(1+\frac{1}{\beta_C\delta^K}\right)\right)\right\rceil + K \\
&\leq \frac{4}{3}\left\lceil \frac{72\|M\|^2}{\rho(M)^2\beta_C^2}\ln\left(\frac{8\|M\|^2}{\rho(M)^2\tau_C}\left(1+\frac{4\|M\|}{\rho(M)\beta_C}\right)\right)\right\rceil + \left\lceil \log_2\left(\frac{\|M\|}{\rho(M)}\right)\right\rceil + 2 \\
&\leq \frac{4}{3}\left\lceil \frac{72\|M\|^2}{\rho(M)^2\beta_C^2}\ln\left(\frac{40\|M\|^3}{\rho(M)^3\tau_C\beta_C}\right)\right\rceil + \left\lceil \log_2\left(\frac{\|M\|}{\rho(M)}\right)\right\rceil + 2 \\
&\leq \frac{4}{3}\left\lceil \frac{216\|M\|^2}{\rho(M)^2\beta_C^2}\ln\left(\frac{40\|M\|}{\rho(M)\tau_C\beta_C}\right)\right\rceil + \left\lceil \log_2\left(\frac{\|M\|}{\rho(M)}\right)\right\rceil + 2.
\end{aligned} \tag{62}$$

The second inequality in (62) follows from (60). We have thus established the second claim of the theorem.

It remains to show that the vector w returned by algorithm HCE satisfies conditions *1* through *4*. Let δ^K denote the value of δ during the last iteration of HCE. Applying Proposition 8 combined with (60) we conclude that conditions *1* and *2* are satisfied. Furthermore,

$$\|w\| \le \frac{1}{2}\delta^K\tau_C + \frac{1}{\beta_C} + \delta^K \le \frac{3}{2} + \frac{1}{\beta_C} \le \frac{5}{2\beta_C},$$

which establishes condition *3*, and

$$\frac{\|w\|}{\text{dist}(w, \partial C)} \le \frac{\frac{1}{2}\delta^K\tau_C + \frac{1}{\beta_C} + \delta^K}{\frac{1}{2}\tau_C\delta^K} = 2\left(\frac{1}{2} + \frac{1}{\beta_C\tau_C\delta^K} + \frac{1}{\tau_C}\right)$$

$$\le 2\left(\frac{1}{2} + \frac{4\|M\|}{\rho(M)\beta_C\tau_C} + \frac{1}{\tau_C}\right) \le \frac{11\|M\|}{\rho(M)\beta_C\tau_C},$$

which establishes condition *4* and completes the proof of the theorem.

□

5. Algorithm CLS for resolving a general conic linear system

In this section we indicate how algorithms HCI and HCE can be used to obtain reliable solutions of a conic linear system in the most general form. A general conic linear system has the form

$$\begin{aligned} (\text{FP}_d)\ & Ax = b \\ & x \in C_X \end{aligned}$$

of (1), and the "strong alternative" system of (FP_d) is

$$\begin{aligned} (\text{SA}_d)\ & A^t s \in C_X^* \\ & b^t s < 0 \end{aligned}$$

of (11). We develop algorithm CLS, which is a combination of two other algorithms, namely algorithm FCLS (Feasible Conic Linear System) which is used to find a reliable solution of (FP_d), and algorithm ICLS (Infeasible Conic Linear System), which is used to find a reliable solution to the alternative system (SA_d). We first proceed by presenting algorithms FCLS and ICLS, and studying their complexity. We then combine algorithms FCLS and ICLS to form algorithm CLS and study its complexity.

Recall that Assumption 1 is presumed to be valid.

5.1. Algorithm FCLS

Algorithm FCLS is designed to compute a reliable solution of (FP_d) of (1) when the system (FP_d) is feasible. Consider the following reformulation of the system (FP_d):

$$\begin{aligned} & -b\theta + Ax = 0 \\ & \theta \ge 0,\ x \in C_X. \end{aligned} \tag{63}$$

System (63) is of the form (HCE) of (40) under the following assignments:

- $M = \begin{bmatrix} -b & A \end{bmatrix}$
- $C = \Re_+ \times C_X$,

with norms defined as follows:

- $\|(\theta, x)\| = |\theta| + \|x\|,\ (\theta, x) \in \Re \times X$
- $\|v\| = \|v\|_2,\ v \in Y$.

Then the norm approximation vector for C is easily seen to be $\bar{u} = (1,\ \bar{f})$ with $\beta_C = \beta$. Moreover, the width of the cone C is $\tau_C = \frac{\tau}{1+\tau} \geq \frac{1}{2}\tau$ and is attained at $u = \frac{1}{1+\tau}(\tau,\ f)$.

Proposition 9. *Suppose (FP_d) of (1) is feasible and $\rho(d) > 0$. Then the system (63) is feasible, M has full rank, and we have*

$$\|M\| = \|d\|,\ \textit{and}\ \rho(M) = \rho(d),$$

where $\rho(M)$ is defined in (46).

Proof. Feasibility of the system (63) is trivially obvious. The expression for $\|M\| = \|d\|$ follows from the definition of the operator norm. The last statement of the proposition is a slightly altered restatement of Theorem 3.5 of [30]. Since $\rho(M) = \rho(d) > 0$, Remark 4 implies that M has full rank.

□

We use algorithm HCE to find a sufficiently interior solution of the system (63) and transform its output into a reliable solution of (FP_d), as described below:

Algorithm FCLS

- *Data:* $d = (A, b)$

  Step 1 Apply algorithm HCE to the system (63). The algorithm will return a vector $\tilde{w} = (\tilde{\theta},\ \tilde{x})$.

  Step 2 Define $\hat{x} = \frac{\tilde{x}}{\tilde{\theta}}$. Return $\hat{x}$ (a reliable solution of (FP_d)).

Lemma 5. *Suppose (FP_d) is feasible and $\rho(d) > 0$. Then algorithm FCLS will terminate in at most*

$$\frac{4}{3}\left\lceil \frac{216\mathcal{C}(d)^2}{\beta^2} \ln\left(\frac{80\mathcal{C}(d)}{\tau\beta}\right)\right\rceil + \lceil \log_2 \mathcal{C}(d)\rceil + 2 \tag{64}$$

iterations of algorithm GVNA. The output $\hat{x}$ will satisfy

1. $\hat{x} \in X_d$,
2. $\|\hat{x}\| \leq \frac{22\mathcal{C}(d)}{\beta\tau} - 1$,

3. $\text{dist}(\hat{x}, \partial C_X) \geq \frac{\beta\tau}{22\mathcal{C}(d)}$,
4. $\frac{\|\hat{x}\|}{\text{dist}(\hat{x}, \partial C_X)} \leq \frac{22\mathcal{C}(d)}{\beta\tau}$.

Proof. To simplify the expressions in this proof, define $\alpha \stackrel{\triangle}{=} \text{dist}(\tilde{w}, \partial C) = \text{dist}\Big((\tilde{\theta}, \tilde{x}), \partial(\Re_+ \times C_X)\Big)$.

From Theorem 2 we conclude that algorithm HCE in Step 1 will terminate in at most

$$\frac{4}{3}\left\lceil \frac{216\mathcal{C}(d)^2}{\beta^2} \ln\left(\frac{80\mathcal{C}(d)}{\tau\beta}\right)\right\rceil + \lceil \log_2 \mathcal{C}(d)\rceil + 2$$

iterations of algorithm GVNA, which establishes the first statement of the lemma.

Next, from Theorem 2 we conclude that the vector $\tilde{w} = (\tilde{\theta}, \tilde{x})$ returned by algorithm HCE in Step 1 satisfies:

$$-b\tilde{\theta} + A\tilde{x} = 0,\ (\tilde{\theta}, \tilde{x}) \in \Re_+ \times C_X,\ \alpha \geq \frac{\tau_C \rho(M)}{8\|M\|} \geq \frac{\tau}{16\mathcal{C}(d)}, \tag{65}$$

$$|\tilde{\theta}| + \|\tilde{x}\| \leq \frac{5}{2\beta_C} = \frac{5}{2\beta},\ \ \frac{\|(\tilde{\theta}, \tilde{x})\|}{\alpha} \leq \frac{11\|M\|}{\rho(M)\beta_C\tau_C} \leq \frac{22\mathcal{C}(d)}{\beta\tau}. \tag{66}$$

Note in particular that (65) implies that $\tilde{\theta} \geq \alpha > 0$, so that $\hat{x}$ is well-defined, and $A\hat{x} = b,\ \hat{x} \in C_X$, which establishes statement *1*.

Next,

$$\|\hat{x}\| = \frac{\|\tilde{x}\|}{\tilde{\theta}} = \frac{\|\tilde{w}\| - \tilde{\theta}}{\tilde{\theta}} \leq \frac{\|\tilde{w}\|}{\alpha} - 1 \leq \frac{22\mathcal{C}(d)}{\beta\tau} - 1,$$

which proves *2*.

To prove *3*, define $t \stackrel{\triangle}{=} \frac{\alpha}{\|\tilde{w}\|}(1 + \|\hat{x}\|)$. Then a simple application of (66) implies that $t \geq \frac{\beta\tau}{22\mathcal{C}(d)}$. Further, let $p \in X$ be an arbitrary vector satisfying $\|p\| \leq t$. Then

$$\|\tilde{\theta}p\| \leq \tilde{\theta} \cdot t = \tilde{\theta} \cdot \frac{\alpha}{\|\tilde{w}\|}(1 + \|\hat{x}\|) = \frac{\alpha}{\|\tilde{w}\|}(\tilde{\theta} + \|\tilde{x}\|) = \alpha,$$

and so $\tilde{x} + \tilde{\theta}p \in C_X$, and hence $\hat{x} + p = \frac{\tilde{x}+\tilde{\theta}p}{\tilde{\theta}} \in C_X$. Therefore, $\text{dist}(\hat{x}, \partial C_X) \geq t \geq \frac{\beta\tau}{22\mathcal{C}(d)}$, establishing *3*.

Finally,

$$\frac{\|\hat{x}\|}{\text{dist}(\hat{x}, \partial C_X)} \leq \frac{\|\hat{x}\|}{t} = \frac{\|\hat{x}\| \cdot \|\tilde{w}\|}{\alpha(1 + \|\hat{x}\|)} \leq \frac{\|\tilde{w}\|}{\alpha} \leq \frac{22\mathcal{C}(d)}{\beta\tau},$$

which establishes *4*.

□

5.2. *Algorithm ICLS*

Algorithm ICLS is designed to compute a reliable solution of (SA_d) of (11) when the system (FP_d) is infeasible. Consider the following compact-form reformulation of the system (FP_d):

$$\begin{array}{c} -b\theta + Ax = 0 \\ \theta + \bar{f}^t x = 1 \\ \theta \geq 0,\ x \in C_X. \end{array} \tag{67}$$

The alternative system to (67) is given by

$$\begin{array}{c} -b^t s > 0 \\ A^t s \in \operatorname{int} C_X^*. \end{array} \tag{68}$$

System (68) is of the form (HCI) under the following assignments:

- $M = \begin{bmatrix} -b & A \end{bmatrix}$
- $C = \Re_+ \times C_X$,

with norms defined as follows:

- $\|(\theta, x)\| = |\theta| + \|x\|,\ (\theta, x) \in \Re \times X$
- $\|v\| = \|v\|_2,\ v \in Y$.

Then the norm approximation vector for C is easily seen to be $\bar{u} = (1,\ \bar{f})$ with $\beta_C = \beta$.

Proposition 10. *Suppose the system (FP_d) is infeasible and $\rho(d) > 0$. Then the system (67) is infeasible, and we have*

$$\|M\| = \|d\|,$$

$$\rho(d) \leq r(M, 0) \leq \frac{\rho(d)}{\beta},$$

where $r(M, 0)$ is defined in (42).

Proof. Suppose the system (67) has a solution $(\tilde{\theta}, \tilde{x})$. Since the system ($FP_d$) is infeasible, we must have $\tilde{\theta} = 0$. Then the perturbed data vector $d + \Delta d = (A + \epsilon b \bar{f}^t, b)$ where $\epsilon > 0$ gives rise to the system ($FP_{d+\Delta d}$) which has a solution $\tilde{x}/\epsilon$. The size of the perturbation $\|\Delta A, \Delta b\| = \|\epsilon b \bar{f}^t, 0\| = \epsilon \|b\|$ can be made arbitrarily small. This indicates that the system (FP_d) is ill-posed, contradicting the assumptions of the proposition. Thus, the system (67) has no solution.

The expression for $\|M\| = \|d\|$ follows from the definition of the operator norm. Next we establish the bounds on $r(M, 0)$. Since the system (67) is infeasible $r(M, 0)$ is computed as

$$\begin{array}{lll} r(M,0) = \min \|0 - M(\theta, x)\| & = \min \|b\theta - Ax\| & \\ \quad \theta + \bar{f}^t x = 1 & \quad \theta + \bar{f}^t x = 1 & \\ \quad \theta \geq 0,\ x \in C_X & \quad \theta \geq 0,\ x \in C_X, & \end{array} \tag{69}$$

which is exactly program $P_g(d)$ of [13] (for the case when $C_Y = \{0\}$). Therefore, applying Theorem 13 of [13] we conclude that $\beta r(M, 0) \leq \rho(d) \leq r(M, 0)$, that is, $\rho(d) \leq r(M, 0) \leq \frac{\rho(d)}{\beta}$.

□

We use algorithm HCI to compute a sufficiently interior solution of the system (68) and show that it is a reliable solution of (SA_d), as described below:

Algorithm ICLS

– *Data:* $d = (A, b)$
  Step 1 Apply algorithm HCI to the system (68). The algorithm will return a vector s.
  Step 2 Return s (a reliable solution of (SA_d)).

Lemma 6. *Suppose (FP_d) is infeasible and $\rho(d) > 0$. Then algorithm ICLS will terminate in at most*

$$\left\lfloor \frac{16\mathcal{C}(d)^2}{\beta^2} \right\rfloor \tag{70}$$

iterations of GVNA. The output s satisfies $s \in A_d$ and

$$\frac{\|s\|_*}{\text{dist}(s, \partial A_d)} \leq \frac{2\mathcal{C}(d)}{\beta}.$$

Proof. From Theorem 1 we conclude that algorithm HCI in Step 1 will terminate in at most

$$\left\lfloor \frac{16\|M\|^2}{\beta_C^2 r(M,0)^2} \right\rfloor \leq \left\lfloor \frac{16\mathcal{C}(d)^2}{\beta^2} \right\rfloor$$

iterations of GVNA, which establishes the first statement of the lemma. Furthermore, the output s satisfies $s \in S_M$ and

$$\frac{\|s\|_*}{\text{dist}(s, \partial S_M)} \leq \frac{2\|M\|}{\beta_C r(M,0)} \leq \frac{2\mathcal{C}(d)}{\beta}.$$

Since $S_M \subseteq A_d$, the result follows.

□

5.3. Algorithm CLS

Algorithm CLS described below is a combination of algorithms FCLS and ICLS. Algorithm CLS is designed to solve the system (FP_d) of (1) by either finding a reliable solution of (FP_d) or demonstrating the infeasibility of (FP_d) by finding a reliable solution of (SA_d). Since it is not known in advance whether (FP_d) is feasible or not, algorithm CLS runs both algorithms FCLS and ICLS in parallel, and terminates when either one of the two algorithms terminates. The formal description of algorithm CLS is as follows:

Algorithm CLS

– *Data:* $d = (A, b)$
  Step 1 Run algorithms FCLS and ICLS in parallel on the data set $d = (A, b)$, until one of them terminates.
  Step 2 If algorithm FCLS terminates first, return its output $\hat{x}$. If algorithm ICLS terminates first, return its output s.

Although Step 1 of algorithm CLS calls for algorithms FCLS and ICLS to be run in parallel, there is no necessity for parallel computation *per se*. Observe that both algorithms FCLS and ICLS consist of repetitively calling the algorithm GVNA on a sequence of data instances. A sequential implementation of Step 1 is to run one iteration of algorithm GVNA called by algorithm FCLS, followed by the next iteration of algorithm GVNA called by the algorithm ICLS, etc., until one of the iterations yields the termination of the algorithm.

Combining the complexity results for algorithms FCLS and ICLS from Lemmas 5 and 6 we obtain the following complexity analysis of algorithm CLS:

Theorem 3. *Suppose that $\rho(d) > 0$ and Assumption 1 is satisfied. If the system (FP_d) is feasible, algorithm CLS will terminate in at most*

$$\frac{8}{3}\left\lceil \frac{216\mathcal{C}(d)^2}{\beta^2} \ln\left(\frac{80\mathcal{C}(d)}{\tau\beta}\right)\right\rceil + 2\left\lceil \log_2 \mathcal{C}(d)\right\rceil + 4$$

iterations of GVNA, and will return a reliable solution $\hat{x}$ of (FP_d). That is, $\hat{x}$ will have the following properties:

- $\hat{x} \in X_d$,
- $\|\hat{x}\| \leq \frac{22\mathcal{C}(d)}{\beta\tau} - 1$,
- $\operatorname{dist}(\hat{x}, \partial C_X) \geq \frac{\beta\tau}{22\mathcal{C}(d)}$,
- $\frac{\|\hat{x}\|}{\operatorname{dist}(\hat{x}, \partial C_X)} \leq \frac{22\mathcal{C}(d)}{\beta\tau}$.

If the system (FP_d) is infeasible, algorithm CLS will terminate in at most

$$2\left\lfloor \frac{16\mathcal{C}(d)^2}{\beta^2} \right\rfloor$$

iterations of GVNA, and will return a reliable solution s of (SA_d), thus demonstrating infeasibility of (FP_d). That is, s will satisfy the following properties:

- $s \in A_d$,
- $\frac{\|s\|_*}{\operatorname{dist}(s, \partial A_d)} \leq \frac{2\mathcal{C}(d)}{\beta}$.

Proof. The proof is an immediate consequence of Lemmas 5 and 6. The bounds on the number of iterations of algorithm GVNA in the theorem are precisely double the bounds in the lemmas, due to running algorithms FCLS and ICLS in parallel.

□

6. Discussion

Discussion of complexity bound and work per iteration. Observe that algorithm CLS (as well as algorithms FCLS and ICLS) consists simply of repetitively calling algorithm GVNA on a sequence of data instances (M, g), all with the same matrix $M = [-b\ A]$, and with right-hand side of the form $g = 0$ or $g = -\delta Mu$ for a sequence of values of the parameters δ. Viewed in this light, algorithm CLS is essentially no

more than algorithm GVNA applied to a sequence of data instances all of very similar form. The total workload of algorithm CLS, as presented in Theorem 3, is the total number of iterations of algorithm GVNA called in algorithm CLS. In this perspective, algorithm CLS is "elementary" in that the computation at each inner iteration is not particularly sophisticated, only involving some matrix-vector multiplications and the solution of one conic section optimization problem (CSOP_{C_X}) per iteration of GVNA. Each iteration of algorithm GVNA used in algorithms FCLS and ICLS uses at most $T_{C_X} + O(mn)$ operations, where T_{C_X} is the number of operations needed to solve an instance of (CSOP_{C_X}) and the term $O(mn)$ derives from counting the matrix-vector and vector-vector multiplications. The number of operations required to perform these multiplications can be significantly reduced if the matrices and vectors involved are sparse.

In addition to running algorithm GVNA, algorithm CLS (in particular, algorithm FCLS) computes several projections using formula (52). This computation cannot be considered elementary since it involves the inverse of the square matrix MM^t and requires $O(m^3)$ iterations. However, since the matrix M used by algorithm FCLS is the same in all projection computations, this step of the algorithm can be implemented by computing the projection matrix $P \stackrel{\triangle}{=} I - M^t(MM^t)^{-1}M$ "off-line" (before calling algorithm CLS). Then the projections required by the algorithm FCLS can be computed by means of matrix-vector multiplication. Since algorithm FCLS will perform no more than $O(\ln(\mathcal{C}(d)))$ computations of Euclidean projections (see Theorem 2), the multiplications involving P will not increase the computation time significantly even though P is not likely to have a nice sparsity structure.

A practical elementary algorithm? This paper has positively addressed two theoretical questions regarding elementary algorithms and the condition number $\mathcal{C}(d)$. It remains to be seen if algorithm CLS, or any other elementary algorithm for solving the problem (FP_d), will be competitive in practice with algorithms such as interior-point methods on a suitable class of problems. Each iteration of algorithm CLS will perform only a few operations when the oracle for solving the problem (CSOP) is efficiently implemented, and when the original problem data is sparse. Furthermore, the number of operations performed in each iteration of CLS is less affected by the growing dimension of the problem then it would be for an interior point algorithm. Therefore, a study of the practical performance of algorithm CLS on problem classes involving large, sparse, and well-structured problems may be a topic of future research investigation.

In this vein, recent literature contains both theoretical and practical studies of several algorithms for obtaining approximate solutions of certain structured convex optimization problems that can be also considered elementary in the above sense, and moreover, are of similar nature to the algorithm CLS. See, for example, Grigoriadis and Khachiyan [17, 18] and Villavicencio and Grigoriadis [39], who consider algorithms for block angular resource sharing problems, Plotkin, Shmoys, and Tardos [27] and Karger and Plotkin [20] who consider algorithms for fractional packing problems, and Bienstock [4, 5] and Goldberg et al. [16], where results of computational experiments with these methods are discussed. Similar to algorithm CLS, each iteration of the algorithms above maintains an "admissible" point (i.e., a point that satisfies a pre-specified subset of problem constraints) and consists of a call to an oracle to solve a linear optimization

subproblem similar to (CSOP), and uses the oracle output to generate a direction and a new iterate with reduced violations in the remaining constraints. In addition to calls to the oracle, most computations performed at each iteration consist of matrix-vector multiplications involving the original data. Most recently, Ben-Tal, Margalit, and Nemirovski [3] also use an elementary algorithm (the general mirror descent scheme) to successfully solve very-large-scale image reconstruction problems.

The many applications of the problems considered in the aforementioned papers include network design problems, multi-commodity network flows, scheduling, combinatorial optimization, image reconstruction, etc. The dimensionality of such structured problems arising in practice is often prohibitively large for theoretically efficient algorithms such as interior-point methods to be effective. However, the computational experience with the above elementary algorithms has shown that elementary algorithms can be a superior alternative (see, in particular, [5] and [3]). The complexity analysis as well as the practical computational experience of this body of literature lends more credence to the practical viability of elementary algorithms in general, when applied to large-scale, sparse, well-structured, and well-conditioned problems.

Other formats of conic linear systems. In this paper, we have assumed that the problem (FP_d) has "primal standard form" $Ax = b,\ x \in C_X$, where C_X is a regular cone. Instead, one might want to consider problems in "standard dual form" $b - Ax \in C_Y,\ x \in X$, or the most general form $b - Ax \in C_Y,\ x \in C_X$. Elementary algorithms for problems in these forms, with the cones C_Y and/or C_X assumed to be regular, are addressed in detail in [9]. In general, these problems can be approached by converting them into primal standard form above and applying algorithm CLS as described in this paper. The technique for converting problems of general form $b - Ax \in C_Y,\ x \in C_X$ into primal standard form was originally suggested by Peña and Renegar [26] and can be interpreted as introducing scaled slack variables for the linear constraints. This approach is extended to problems in standard dual form in [9]. In some cases, however, the problem can be treated by an elementary algorithm directly, without converting it into standard form. These approaches are also presented in detail in [9].

Converting Algorithm CLS into an optimization algorithm. Converting algorithm CLS into an optimization algorithm is a logical extension of the work presented in this paper. Suppose that we are interested in minimizing a linear function $c^t x$ over the feasible region of (FP_d). Then algorithm CLS could be modified, for example, with the addition of an outer loop that will add an objective function cut of the form $c^t x \leq c^t \bar{x}$ whenever a solution $\bar{x}$ is produced at the previous iteration. This may be a topic of future research.

Ill-posed problem instances. The complexity bound of Theorem 3 relies on the fact that (FP_d) is not ill-posed, i.e., $\rho(d) > 0$. The algorithm CLS is not predicted to perform well (and in fact, is not guaranteed to terminate) in cases when $\rho(d) = 0$. This does not constitute, in our view, a weakness of the algorithm, since such problems are exceptionally badly behaved in general. In particular, an arbitrarily small perturbation of the data can change the feasibility status of such problems, which makes it rather hopeless to compute exact solutions or certificates of infeasibility.

References

1. Agmon, S. (1954): The relaxation method for linear inequalities. Can. J. Math. **6**, 382–392
2. Alizadeh, F. (1995): Interior point methods in semidefinite programming with applications to combinatorial optimization. SIAM J. Optim. **5**(1), 13–51
3. Ben-Tal, A., Margalit, T., Nemirovski, A. (2000): The ordered subsets mirror descent optimization method and its use for the positron emission tomography reconstruction problem. Technical Report, MINERVA Optimization Center, Technion – Israel Institute of Technology, Haifa, Israel
4. Bienstock, D. (1996): Experiments with a network design algorithm using ϵ-approximate linear programs. Technical Report 1999-4. CORC, Columbia Univeristy
5. Bienstock, D. (1999): Approximately solving large-scale linear programs. I. Strengthening lower bounds and accelerating convergence. Technical Report 1999-1. CORC, Columbia Univeristy
6. Dantzig, G.B. (1991): Converting a converging algorithm into a polynomially bounded algorithm. Technical Report SOL 91-5. Stanford University
7. Dantzig, G.B. (1992): An ϵ-precise feasible solution to a linear program with a convexity constraint in $1/\epsilon^2$ iterations independent of problem size. Technical Report. Stanford University
8. Eaves, B.C. (1973): Piecewise linear retractions by reflection. Linear Algebra Appl. **7**, 93–98
9. Epelman, M. (1999): Complexity, Condition Numbers, and Conic Linear Systems. PhD thesis. Massachusetts Institute of Technology
10. Epelman, M., Freund, R.M. (1997): Condition number complexity of an elementary algorithm for resolving a conic linear system. Working Paper OR 319-97. Operations Research Center, Massachusetts Institute of Technology
11. Filipowski, S. (1994): On the complexity of solving linear programs specified with approximate data and known to be feasible. Technical Report. Dept. of Industrial and Manufacturing Systems Engineering, Iowa State University
12. Filipowski, S. (1994): On the complexity of solving sparse symmetric linear programs specified with approximate data. Technical Report. Dept. of Industrial and Manufacturing Systems Engineering, Iowa State University
13. Freund, R.M., Vera, J.R. (1999): Some characterizations and properties of the "distance to ill-posedness" and the condition measure of a conic linear system. Math. Program. **86**(2), 225–260
14. Freund, R.M., Vera, J.R. (2000): Condition-based complexity of convex optimization in conic linear form via the ellipsoid algorithm. SIAM J. Optim. **10**(1), 155–176
15. Goffin, J.L. (1980): The relaxation method for solving systems of linear inequalities. Math. Oper. Res. **5**(3), 388–414
16. Goldberg, A.V., Oldham, J.D., Plotkin, S., Stein, C. (1998): An implementation of a combinatorial approximation algorithm for minimum-cost multicommodity flow. Technical Report 98-038. NEC Research Institute, Inc.
17. Grigoriadis, M.D., Khachiyan, L.G. (1994): Fast approximation schemes for convex programs with many blocks and coupling constraints. SIAM J. Optim. **4**(1), 86–107
18. Grigoriadis, M.D., Khachiyan, L.G. (1996): Coordination complexity of parallel price-directive decomposition. Math. Oper. Res. **21**(2), 321–340
19. Horn, R.A., Johnson, C.R. (1985): Matrix Analysis. Cambridge University Press, New York
20. Karger, D., Plotkin, S. (1995): Adding multiple cost constraints to combinatorial optimization problems, with applications to multicommodity flows. Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing, pp. 18–25
21. Motzkin, T.S., Schoenberg, I.J. (1954): The relaxation method for linear inequalities. Can. J. Math. **6**, 393–404
22. Nesterov, Y., Nemirovskii, A. (1994): Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics (SIAM), Philadelphia
23. Nunez, M.A., Freund, R.M. (1998): Condition measures and properties of the central trajectory of a linear program. Math. Program. **83**(1), 1–28
24. Peña, J. (1997): Computing the distance to infeasibility: theoretical and practical issues. Technical Report. Cornell University
25. Peña, J. (2000): Understanding the geometry of infeasible perturbations of a conic linear system. SIAM J. Optim. **10**(2), 534–550
26. Peña, J., Renegar, J. (2000): Computing approximate solutions for convex conic systems of constraints. Math. Program. DOI 10.1007/s101070000136
27. Plotkin, S.A., Shmoys, D.B., Tardos, É. (1995): Fast approximation algorithms for fractional packing and covering problems. Math. Oper. Res. **20**(2), 257–301
28. Renegar, J. (1994): Some perturbation theory for linear programming. Math. Program. **65**(1), 73–91

29. Renegar, J. (1995): Incorporating condition measures into the complexity theory of linear programming. SIAM J. Optim. **5**(3), 506–524
30. Renegar, J. (1995): Linear programming, complexity theory, and elementary functional analysis. Math. Program. **70**(3), 279–351
31. Rosenblatt, F. (1958): The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Rev. **65**, 386–408
32. Rosenblatt, F. (1960): On the convergence of reinforcement procedures in simple perceptrons. Report. VG-1196-G-4. Cornell Aeronautical Laboratory, Buffalo, NY
33. Rosenblatt, F. (1960): Perceptron simulation experiments. Proc. Inst. Radio Eng. **48**, 301–309
34. Rosenblatt, F. (1962): Principles of Neurodynamics. Spartan Books, Washington, DC
35. Vera, J.R. (1992): Ill-posedness and the computation of solutions to linear programs with approximate data. Technical Report. Cornell University
36. Vera, J.R. (1992): Ill-Posedness in Mathematical Programming and Problem Solving with Approximate Data. PhD thesis. Cornell University
37. Vera, J.R. (1996): Ill-posedness and the complexity of deciding existence of solutions to linear programs. SIAM J. Optim. **6**(3), 549–569
38. Vera, J.R. (1998): On the complexity of linear programming under finite precision arithmetic. Math. Program. **80**(1), 91–123
39. Villavicencio, J., Grigoriadis, M.D. (1997): Approximate Lagrangian decomposition with a modified Karmarkar logarithmic potential. In: Network optimization (Gainesville, FL, 1996), pp. 471–485. Springer, Berlin