OPERATIONS RESEARCH CENTER

Working Paper

Condition Number Complexity of an Elementary Algorithm for Computing a Reliable Solution of a Conic Linear System.

by
M. Epelman
R. Freund

OR 330-98 December 1998

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Condition Number Complexity of an Elementary Algorithm for Computing a Reliable Solution of a Conic Linear System[1]

Marina Epelman,[2] M.I.T. Robert M. Freund,[3] M.I.T.

December, 1998

Abstract

A conic linear system is a system of the form

$$(\mathrm{FP}_d): \quad \begin{array}{l} Ax = b \\ x \in C_X, \end{array}$$

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., lies in the intersection of the closures of 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 a scale-invariant reciprocal of $\rho(d)$: $\mathcal{C}(d) \triangleq \frac{\|d\|}{\rho(d)}$.

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

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

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 mathematical operations at each iterations.

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).

AMS Subject Classification: 90C, 90C05, 90C60

Keywords: Complexity of Convex Programming, Conditioning, Error Analysis

---

[1]This research has been partially supported through NSF Graduate Research Fellowship.

[2]M.I.T. Operations Research Center, Bldg. E40-130, 77 Massachusetts Avenue, Cambridge, MA 02139-4307. e-mail: mepelman@mit.edu

[3]M.I.T. Operations Research Center, Bldg. E40-111, 77 Massachusetts Avenue, Cambridge, MA 02139-4307. email: rfreund@mit.edu

1 Introduction

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

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

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 [21] and Renegar [28] and [29], among others, wherein interior-point methods for (FP_d) are shown to be theoretically efficient.

Our interest lies in instances of (FP_d) where an interior-point or other theoretically-efficient algorithm may not be an attractive choice for solving (FP_d). Such instances might arise when n is extremely large, and/or when A is a real matrix whose sparsity structure is incompatible with efficient computation in interior-point methods, for example.

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 (i.e., 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 of von Neumann studied by Dantzig [5] and [6], 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], [20], [7], [14]), the "perceptron" algorithm [30, 31, 32, 33], 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. Also, the performance of these algorithms can be quite competitive when applied to certain very large problems with very sparse data, see [4]. We refer to these algorithms as "elementary" in that the algorithms do not involve particularly sophisticated mathematics at each iteration, nor do the algorithms 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 [27, 28, 29]. $\mathcal{C}(d)$ 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 [27], certain geometric properties of X_d [12], and the complexity of algorithms for computing solutions of (FP_d) [29], [13]. (The concepts underlying $\mathcal{C}(d)$ will be reviewed in detail at the end of this section.) We show in Section 5 that algorithm CLS will compute a feasible solution of (FP_d) in

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

iterations when (FP_d) is feasible, or will demonstrate infeasibility in

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

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, $\text{dist}(\hat{x}, \partial C_X)$, 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 $\text{dist}(\hat{x}, \partial C_X)$ 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 [12] 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 lends credence to 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 [29], Renegar presented an incredibly general interior-point (i.e., barrier) algorithm for resolving (FP_d) and showed, roughly speaking, that the iteration complexity bound of the algorithm depends linearly and only on two quantities: the barrier parameter for the cone C_X, and $\ln(\mathcal{C}(d))$, i.e., the logarithm of the condition number $\mathcal{C}(d)$. In [13] several efficient volume-reducing cutting-plane algorithms for resolving (FP_d) (such as the ellipsoid algorithm) are shown to have iteration complexity that is linear in $\ln(\mathcal{C}(d))$ and polynomial in the dimension n. 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 [28]. Both the interior-point algorithm and the ellipsoid algorithm are also very sophisticated algorithms, in contrast with 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 Section 2) and does not perform any sophisticated mathematics at each iteration. Consequently one would not expect its theoretical complexity to be nearly as good as an interior-point algorithm or the ellipsoid algorithm. However, because the work per iteration of algorithm CLS is low, and each iteration fully exploits the sparsity of the original data, it is reasonable to expect that algorithm CLS could outperform more theoretically-efficient algorithms on large structured problems that are well-conditioned.

In this vein, recent literature contains several algorithms of similar nature to the elementary algorithms discussed above, for obtaining approximate solutions of certain structured convex optimization problems. For example, Grigoriadis and Khachiyan [16, 17] and Villavicencio and Grigoriadis [38] consider algorithms for block angular resource sharing problems, Plotkin, Shmoys, and Tardos [26] and Karger and Plotkin [19] consider algorithms for fractional packing problems, and Bienstock [3] and Goldberg et al. [15] discuss results of computational experiments with these methods. The many applications of such problems include multi-commodity network flows, scheduling, combinatorial optimization, 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, these problems are typically sparse and structured, which allows for efficient implementation and good performance of Lagrangian-decomposition based algorithms (see, for example, [38]), which offer a general framework for row-action methods. These algorithms can also be considered "elementary" in the exact same sense as the row-action algorithms mentioned earlier, i.e., they do not perform any sophisticated mathematics at each iteration and they fully exploit the sparsity of the original data. 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.

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 12, 15 and 16. 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 Section 5, we present algorithm CLS for the conic linear system (FP_d). Algorithm CLS is a combination of algorithms HCI and HCE. Theorem 28 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

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

Then $\mathcal{F}$ corresponds to those data instances $d = (A, b)$ for which (FP_d) is feasible. The complement of $\mathcal{F}$ is denoted by $\mathcal{F}^C$, and so $\mathcal{F}^C$ consists precisely of those data instances $d = (A, b)$ for which (FP_d) is infeasible.

The boundary of $\mathcal{F}$ and of $\mathcal{F}^C$ is precisely the set

$$\mathcal{B} = \partial\mathcal{F} = \partial\mathcal{F}^C = \text{cl}(\mathcal{F}) \cap \text{cl}(\mathcal{F}^C) \tag{8}$$

where ∂S denotes the boundary and $\text{cl}(S)$ denotes the closure of a set S. Note that if $d = (A, b) \in \mathcal{B}$, then ($\text{FP}_d$) 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:

$$\rho(d) = \inf\{\|\Delta d\| : d + \Delta d \in \mathcal{B}\},$$

see [27], [28], [29], and so $\rho(d)$ is the distance of the data instance $d = (A, b)$ to the set $\mathcal{B}$ of ill-posed instances for the problem (FP_d). It is straightforward to show that

$$\rho(d) = \begin{cases} \inf\{\|d - \bar{d}\| : \bar{d} \in \mathcal{F}^C\} & \text{if } d \in \mathcal{F}, \\ \inf\{\|d - \bar{d}\| : \bar{d} \in \mathcal{F}\} & \text{if } d \in \mathcal{F}^C, \end{cases} \tag{9}$$

so that we could also define $\rho(d)$ by employing (9). The *condition number* $\mathcal{C}(d)$ of the data instance d is defined to be:

$$\mathcal{C}(d) = \frac{\|d\|}{\rho(d)} \tag{10}$$

when $\rho(d) > 0$, and $\mathcal{C}(d) = \infty$ when $\rho(d) = 0$. The condition number $\mathcal{C}(d)$ 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 $\mathcal{C}(d) \geq 1$. The value of $\mathcal{C}(d)$ is a measure of the relative conditioning of the data instance d. Further analysis of the distance to ill-posedness has been presented in [12], Vera [34, 35, 37, 36], Filipowski [10, 11], Nunez and Freund [22], Peña [23, 24] and Peña and Renegar [25].

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 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.

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.*

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

The "strong alternative" system of (FP_d) is:

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

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

Proposition 2 *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 3 *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 (11) 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 [12] it is shown that when the system (FP_d) is infeasible, there exists a point $\tilde{s} \in A_d$ such that

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

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 shortly.) Algorithm CLS will compute a solution $\hat{s}$ with a bound of the same order as (13).

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:

$$\|z\|_* = \max\{z^t x : \|x\| \leq 1\}, \tag{14}$$

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 $\|A\| = \|v\|_* \|u\|$.

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 4 *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 5 *If C is a regular cone in the normed linear vector space X, the coefficient of linearity for the cone C is given by:*

$$\beta_C = \begin{array}{ll} \sup & \inf u^T x \\ u \in X^* & x \in C \\ \|u\|_* = 1 & \|x\| = 1. \end{array} \tag{15}$$

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 6 (see [12]) $0 < \beta_C \leq 1$. *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 6 we refer to $\bar{u}$ as the norm linearization vector for the cone C. The following proposition shows that the width of C is equal to the coefficient of linearity for C^*:

Proposition 7 (see [13]) *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 $C = \Re^n_+ \triangleq \{x \in \Re^n : x \geq 0\}$. 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 for $x \in C$ 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 $x = e$, 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 [21]). 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 $C = S^{k\times k}_+ \triangleq \left\{x \in S^{k\times k} : x \succeq 0\right\}$, 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 $C^* = S^{k\times k}_+$, i.e., C is self-dual. For $x \in X$, let $\lambda(x)$ denote the k-vector of ordered eigenvalues of x. For any $p \in [1, \infty)$, 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 [18], 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 $x = I$ 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:

$$\begin{array}{llll} (\mathrm{CSOP}_C) & \min\limits_{x} & c^t x & \\ & \text{s.t.} & x \in C & \\ & & \bar{u}^t x = 1. & \end{array} \tag{16}$$

Let T_C denote an upper bound on the number of operations needed to solve (CSOP_C).

For the algorithm CLS developed in this paper, we presume that we can work conveniently with the cone C_X in that we can solve (CSOP_{C_X}) easily, i.e., that T_{C_X} is not excessive, for otherwise the algorithm will not be very efficient.

We now pause to illustrate how (CSOP_C) is easily solved 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 $i \in \operatorname{argmin}\{c_j : j = 1, \ldots, n\}$. 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 $\|x\| = \|(v, w)\| \triangleq \|v\| + \|w\|$, 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 lemmas which give a precise mathematical characterization of the problem of computing the distance from a given point to the boundary of a given convex set. 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 defined as:

$$r = \min_z \{\|f - z\| : z \in \partial S\}. \tag{17}$$

Lemma 8 *Let r be defined by (17). Suppose $f \in S$. Then*

$$\begin{array}{rccl} r = & \min\limits_{v} & \max\limits_{z} & \theta \\ & \|v\| \le 1 & s.t. & f - z - \theta v = 0 \\ & & & z \in S. \end{array}$$

Lemma 9 *Let r be defined by (17). Suppose $f \notin S$. Then*

$$\begin{array}{rcl} r = & \min\limits_{z} & \|f - z\| \\ & s.t. & z \in S. \end{array}$$

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 [5] and [6], see also [9]. We will work with a conic linear system of the form:

$$\begin{array}{rr} \text{(P)} & Mx = g \\ & x \in C \\ & \bar{u}^t x = 1, \end{array} \tag{18}$$

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 6 is known and given. (The original algorithm of von Neumann presented and analyzed by Dantzig in [5] and [6] was developed for the case when $C = \Re^n_+$ and $\bar{u} = e$.) We will refer to a system of the form (18) as a conic linear system in compact form, or simply a compact-form system.

The "alternative" system to (P) of (18) is:

$$\text{(A)} \quad M^t s - \bar{u}(g^t s) \in \text{int} C^*, \tag{19}$$

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

Proposition 10 *Exactly one of the systems (P) of (18) and (A) of (19) has a solution.*

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

$$\begin{array}{lll}(\mathrm{OP}) & \min\limits_{x} & \|g - Mx\| \\ & \text{s.t.} & x \in C \\ & & \bar{u}^t x = 1.\end{array}$$

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 (16):

$$\begin{array}{lll} \min\limits_{p} & \bar{v}^t(g - Mp) = & \min\limits_{p} \quad \bar{v}^t(g\bar{u}^t - M)p \\ \text{s.t.} & p \in C & \text{s.t.} \quad p \in C \\ & \bar{u}^t p = 1 & \qquad\quad \bar{u}^t p = 1, \end{array} \tag{20}$$

where (20) 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 (20), 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 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

$$\tilde{x}(\lambda) = \bar{x} + \lambda(\bar{p} - \bar{x}).$$

Then the next iterate $\tilde{x}$ is computed as $\tilde{x} = \tilde{x}(\lambda^*)$, where

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

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:

$$\min_{\lambda \in [0,1]} \|(1-\lambda)\bar{v} + \lambda\bar{w}\|^2 = \min_{\lambda \in [0,1]} \lambda^2\|\bar{v} - \bar{w}\|^2 + 2\lambda(\bar{v}^t(\bar{w} - \bar{v})) + \|\bar{v}\|^2. \tag{21}$$

The closed-form solution of the program (21) is easily seen to be

$$\lambda^* = \min\left\{\frac{\bar{v}^t(\bar{v}-\bar{w})}{\|\bar{v}-\bar{w}\|^2}, 1\right\}. \tag{22}$$

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 Solve the following conic section optimization problem (CSOP_C):

$$\begin{array}{llcll} \min\limits_{p} & (v^{k-1})^t(g - Mp) = & & \min\limits_{p} & (v^{k-1})^t(g\bar{u}^t - M)p \\ \text{s.t.} & p \in C & & \text{s.t.} & p \in C \\ & \bar{u}^t p = 1 & & & \bar{u}^t p = 1. \end{array} \tag{23}$$

  Let p^{k-1} be an optimal solution of the optimization problem (23) and $w^{k-1} = g - Mp^{k-1}$. Evaluate STOP$(\cdot)$. If STOP$(\cdot) = 1$, stop, return appropriate output.

  Step 3 Else, let

$$\begin{aligned} \lambda^{k-1} &= \mathrm{argmin}_{\lambda \in [0,1]}\{\|g - M(x^{k-1} + \lambda(p^{k-1} - x^{k-1}))\|\} \\ &= \min\left\{\frac{(v^{k-1})^t(v^{k-1} - w^{k-1})}{\|v^{k-1} - w^{k-1}\|^2}, 1\right\} \end{aligned} \tag{24}$$

  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 and only if $(v^{k-1})^t w^{k-1} > 0$. 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 11). Therefore, algorithm GVNA with STOP = STOP1 will terminate only if the system (P) is infeasible.

2. $\text{STOP2}(v^{k-1}, w^{k-1}) = 1$ if and only if $(v^{k-1})^t w^{k-1} > \frac{\|v^{k-1}\|^2}{2}$. This termination criterion is a stronger version of the previous one.

3. $\text{STOP3}(v^{k-1}, w^{k-1}, k) = 1$ if and only if $(v^{k-1})^t w^{k-1} > 0$ or $k \geq I$, where I is some pre-specified integer. This termination criterion is essentially equivalent to STOP1, but it ensures finite termination (in no more that I iterations) regardless of the status of (P).

Proposition 11 *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 solutions 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 [5] and [6], we regard algorithm GVNA as "elementary" in that the algorithm does not rely on particularly sophisticated mathematics 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 T_C (the number of operations needed to solve (CSOP_C)) is small. A thorough evaluation of the work per iteration of algorithm GVNA is presented in Remark 17 at the end of this section.

As was mentioned in the discussion preceding the statement of the algorithm, the size of the residual $\|v^k\|$ is decreased at each iteration. The rate of decrease depends of 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 [5].

Lemma 12 (Dantzig [5]) *If algorithm GVNA with* STOP = STOP1 *(or* STOP = STOP3*) has performed k (complete) iterations, then*

$$\|v^k\| \leq \frac{\|M - g\bar{u}^t\|}{\beta_C \sqrt{k}}. \tag{25}$$

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

$$\|g - Mx\| = \|(g\bar{u}^t - M)x\| \leq \|M - g\bar{u}^t\| \cdot \|x\| \leq \frac{\|M - g\bar{u}^t\|}{\beta_C}. \tag{26}$$

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 (26).

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

$$\begin{aligned} \|v^k\| = \|g - Mx^k\| &= \|(1-\lambda^{k-1})(g - Mx^{k-1}) + \lambda^{k-1}(g - Mp^{k-1})\| \\ &= \|(1-\lambda^{k-1})v^{k-1} + \lambda^{k-1}w^{k-1}\|, \end{aligned} \tag{27}$$

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 (27). We will substitute $\lambda = \frac{1}{k}$. Making this substitution, we obtain:

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

Squaring (28) yields:

$$\|v^k\|^2 \leq \frac{1}{k^2}\left((k-1)^2\|v^{k-1}\|^2 + \|w^{k-1}\|^2 + 2(k-1)(v^{k-1})^t(w^{k-1})\right). \tag{29}$$

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), $(v^{k-1})^t w^{k-1} \leq 0$. Also, since p^{k-1} is admissible, $\|w^{k-1}\| = \|g - Mp^{k-1}\| \leq \frac{\|M - g\bar{u}^t\|}{\beta_C}$. Combining these results with the inductive bound on $\|v^{k-1}\|$ and substituting into (29) above yields

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

∎

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

$$\mathcal{H} = \mathcal{H}_M = \{Mx : x \in C,\ \bar{u}^t x = 1\}, \tag{30}$$

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

$$r = r(M,g) = \inf\{\|g - h\| : h \in \partial\mathcal{H}\} \tag{31}$$

where $\mathcal{H}$ is defined above in (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.

The following proposition gives a useful characterization of the value of r.

Proposition 13 *Let $\mathcal{H} = \mathcal{H}_M$ and $r = r(M,g)$ be defined as in (30) and (31). If (P) has a feasible solution, then*

$$r = \begin{array}{ccl} \min\limits_{v} & \max\limits_{h} & \theta \\ \|v\| \le 1 & s.t. & g - h - \theta v = 0 \\ & & h \in \mathcal{H} \end{array} = \begin{array}{ccl} \min\limits_{v} & \max\limits_{x} & \theta \\ \|v\| \le 1 & s.t. & g - Mx - \theta v = 0 \\ & & x \in C \\ & & \bar{u}^t x = 1. \end{array} \tag{32}$$

If (P) does not have a feasible solution, then

$$r = \begin{array}{cl} \min\limits_{h} & \|g - h\| \\ s.t. & h \in \mathcal{H} \end{array} = \begin{array}{cl} \min\limits_{x} & \|g - Mx\| \\ s.t. & x \in C \\ & \bar{u}^t x = 1. \end{array} \tag{33}$$

Proof: The proof is a straightforward consequence of Lemmas 8 and 9. ∎

In light of Proposition 13, 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 14 *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 $v^k \neq 0$. 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,

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

Therefore

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

∎

Proposition 14 is used to prove the following linear convergence rate for algorithm GVNA:

Lemma 15 *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*

$$\|v^k\| \le \|v^0\| e^{-\frac{k}{2}\left(\frac{\beta_C r(M,g)}{\|M - g\bar{u}^t\|}\right)^2}. \tag{34}$$

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 $\lambda^* = \min\left\{\frac{\bar{v}^t(\bar{v}-\bar{w})}{\|\bar{v}-\bar{w}\|^2}, 1\right\}$. 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), $\bar{v}^t\bar{w} \le 0$. Therefore

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

so that $\frac{\bar{v}^t(\bar{v}-\bar{w})}{\|\bar{v}-\bar{w}\|^2} \le 1$ and $\lambda^* = \frac{\bar{v}^t(\bar{v}-\bar{w})}{\|\bar{v}-\bar{w}\|^2}$. Substituting this value of λ^* into (35) yields:

$$\|\tilde{v}\|^2 = \frac{\|\bar{v}\|^2\|\bar{w}\|^2 - (\bar{v}^t\bar{w})^2}{\|\bar{v} - \bar{w}\|^2}. \tag{36}$$

Recall from Proposition 14 that $\bar{v}^t\bar{w} \le -r(M,g)\|\bar{v}\|$. Thus, $\|\bar{v}\|^2(\|\bar{w}\|^2 - r(M,g)^2)$ is an upper bound on the numerator of (36). Also, $\|\bar{v} - \bar{w}\|^2 = \|\bar{v}\|^2 + \|\bar{w}\|^2 - 2\bar{v}^t\bar{w} \ge \|\bar{w}\|^2$. Substituting this into (36) yields

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

where the last inequality derives from (26). 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 $\tilde{v} = v^k$,

$$\|v^k\| \leq \|v^{k-1}\| e^{-\frac{1}{2}\left(\frac{\beta_C r(M,g)}{\|g\bar{u}^t - M\|}\right)^2}. \tag{37}$$

Applying (37) 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 16 *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

$$\|\tilde{v}\|^2 = \|(1 - \lambda^*)\bar{v} + \lambda^* \bar{w}\|^2 = (\lambda^*)^2 \|\bar{v} - \bar{w}\|^2 + 2\lambda^* \bar{v}^t(\bar{w} - \bar{v}) + \|\bar{v}\|^2, \tag{38}$$

where λ^* is given by (22). 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 (38), algebraic manipulations yield

$$\|\tilde{v}\|^2 = \|\bar{w}\|^2 \leq \bar{w}^t \bar{v} \leq \frac{\|\bar{v}\|^2}{2} = \|\bar{v}\|^2 - \frac{\|\bar{v}\|^2}{2} \leq \|\bar{v}\|^2 - \frac{\|\bar{v}\|^4 \beta_C^2}{16\|M - g\bar{u}^t\|^2}. \tag{39}$$

The second inequality in (39) follows from the assumption that the algorithm did not terminate at the present iteration. This implies that the termination criterion was not met, i.e., $\bar{v}^t \bar{w} \leq \frac{\|\bar{v}\|^2}{2}$. 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 (38) 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

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

Combining Case 1 and Case 2, we conclude that

$$\|\tilde{v}\|^2 \leq \|\bar{v}\|^2 - \frac{\|\bar{v}\|^4}{\gamma^2}, \text{ where } \gamma \triangleq \frac{4\|M-g\bar{u}^t\|}{\beta_C}. \tag{40}$$

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

$$\|v^k\|^2 \leq \frac{\gamma^2}{k}. \tag{41}$$

First, note that $\|v^1\|^2 \leq \frac{\|M-g\bar{u}^t\|^2}{\beta_C^2} \leq \frac{\gamma^2}{1}$, thus establishing (41) for $k=1$. Suppose that (41) 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 (41), which completes the proof of the lemma. ∎

To complete the analysis of algorithm GVNA, we now discuss the computational work performed per iteration. We have the following remark:

Remark 17 *Each iteration of algorithm GVNA requires at most*

$$T_C + O(mn)$$

operations, where T_C is the number of operations needed to solve an instance of ($CSOP_C$). 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.

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

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

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

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

Here the notation is the same as in Section 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 6 and Proposition 7 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 Section 5 in the development of algorithm CLS for general conic linear system (FP_d).

4.1 Homogeneous Conic Inequality System

In this subsection, we develop algorithm HCI (for Homogeneous Conic Inequalities) and analyze its complexity and the properties of solutions it generates. Algorithm HCI is designed to obtain a solution of the problem

$$\text{(HCI)} \quad M^t s \in \text{int}C^*. \tag{42}$$

We will assume for the rest of this subsection that the system (HCI) of (42) is feasible. We denote the set of solutions of (HCI) by S_M, i.e.,

$$S_M \triangleq \{s : M^t s \in \text{int}C^*\}.$$

The solution s returned by algorithm HCI is "sufficiently interior" in the sense that the ratio $\frac{\|s\|_*}{\text{dist}(s,\partial S_M)}$ 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 (42) is of the form (19) (with $g = 0$). (HCI) is the "alternative" system for the following problem:

$$\text{(PHCI)} \quad \begin{aligned} & Mx = 0 \\ & x \in C \\ & \bar{u}^t x = 1, \end{aligned} \tag{43}$$

which is a system of the form (18). Following (31) we define

$$r(M,0) \triangleq \inf\{\|h\| : h \in \partial\mathcal{H}\}, \tag{44}$$

where, as in (30), $\mathcal{H} \triangleq \{Mx : x \in C,\ \bar{u}^t x = 1\}$. Combining Proposition 13 and a separating hyperplane argument, we easily have the following result:

Proposition 18 *Suppose (HCI) of (42) is feasible. Then (PHCI) of (43) is infeasible and* $r(M,0) = \min\{\|Mx\| :\ x \in C,\ \bar{u}^t x = 1\}$. *Furthermore,* $r(M,0) > 0$.

Algorithm HCI, described 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 $s \triangleq -\frac{\bar{v}}{\|\bar{v}\|}$. Return s.

The following theorem presents an analysis of the iteration complexity of algorithm HCI, and shows that the output s of HCI is a sufficiently interior solution of the system (HCI).

Theorem 19 *Suppose (HCI) is feasible. Algorithm HCI will terminate in at most*

$$\left\lfloor \frac{16\|M\|^2}{\beta_C^2 r(M,0)^2} \right\rfloor \tag{45}$$

iterations of algorithm GVNA.

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

$$\frac{\|s\|}{\text{dist}(s, \partial S_M)} \leq \frac{2\|M\|}{\beta_C r(M,0)}. \tag{46}$$

Proof: Suppose that algorithm GVNA (called in algorithm HCI) has completed k iterations. From Lemma 16 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 18, $r(M,0) \leq \|Mx\|$ 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 $\text{dist}(s, \partial S_M) \geq \frac{\beta_C r(M,0)}{2\|M\|}$. 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

$$\left(M^t\left(s + \frac{\beta_C r(M,0)}{2\|M\|} q\right)\right)^t p = s^t M p + \frac{\beta_C r(M,0)}{2\|M\|} q^t M p. \tag{47}$$

Observe that by definition of s

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

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,

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

Substituting the above two bounds into (47), 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 Homogeneous Conic Equality System

In this subsection, we develop algorithm HCE (for Homogeneous Conic Equalities) and analyze its complexity and the properties of solutions it generates. Algorithm HCE is designed to obtain a solution of the problem

$$\begin{array}{ll}(\text{HCE}) & Mw = 0 \\ & w \in C. \end{array} \tag{48}$$

We assume that M has full rank. We denote the set of solutions of (HCE) by W_M, i.e.,

$$W_M \triangleq \{w : Mw = 0, \; w \in C\}.$$

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) of (48) 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

$$\begin{array}{rccl} \rho(M) \triangleq & \min\limits_{v} & \max\limits_{w} & \theta \\ & \|v\| \le 1 & \text{s.t.} & Mw - \theta v = 0 \\ & & & w \in C \\ & & & \|w\| \le 1. \end{array} \tag{49}$$

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

Remark 20 *Suppose $\rho(M) > 0$. Then the set $\{w \in W_M : w \neq 0\}$ is non-empty, and M has full rank. Moreover, $\rho(M) \le \|M\|$ and*

$$\|(MM^t)^{-1}\| \le \frac{1}{\rho(M)^2}. \tag{50}$$

This follows from the observation that $\rho(M)^2 \le \lambda_1(MM^t)$, 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 20 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 (18), 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

$$(\mathrm{HCE}_0) \quad \begin{aligned} Mx &= 0 \\ x &\in C \\ \bar{u}^t x &= 1. \end{aligned} \tag{51}$$

The system (HCE_0) is derived from the system (HCE) by adding a compactifying constraint $\bar{u}^t x = 1$. Remark 20 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

$$(\mathrm{HCE}_\delta) \quad \begin{aligned} Mx &= \delta z \\ x &\in C \\ \bar{u}^t x &= 1, \end{aligned} \tag{52}$$

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 (31). Recall that the system (HCE_δ) is feasible precisely when $\delta z \in \mathcal{H} \triangleq \{Mx : x \in C,\ \bar{u}^t x = 1\}$. Also, if the system (HCE_δ) 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 21 *Suppose (HCE_0) of (51) is feasible, and $z \in Y$, $z \neq 0$. Define*

$$\bar{\delta} = \max\{\delta : (HCE_\delta) \text{ is feasible}\}. \tag{53}$$

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 (HCE_δ) 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 a closed set, $\bar{\delta}$ is well defined. Note that the definition of $\bar{\delta}$ implies that $\bar{\delta} z \in \partial \mathcal{H}$. Also, since $z \neq 0$ and $\mathcal{H}$ is bounded, $\bar{\delta} < +\infty$. 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 (HCE_δ) 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, we need 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 (52) with the vector $z \triangleq -Mu$, where u is as specified in Assumption 2. The system (HCE$_\delta$) becomes

$$\begin{array}{ll} (\mathrm{HCE}_\delta) & Mx = -\delta M u \\ & x \in C \\ & \bar{u}^t x = 1. \end{array} \tag{54}$$

The following proposition indicates how approximate solutions of the system (HCE$_\delta$) of (54) can be used to obtain sufficiently interior solutions of the system (HCE).

Proposition 22 *Suppose $\rho(M) > 0$ and $\delta > 0$. Suppose further that x is an admissible point for (HCE$_\delta$), and in addition x satisfies*

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

Define

$$w \triangleq (I - M^t(MM^t)^{-1}M)(x + \delta u). \tag{55}$$

Then $Mw = 0$ and

$$\|w - (x + \delta u)\| \leq \frac{1}{2}\delta \tau_C \tag{56}$$

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

Proof: First, observe that w satisfies $Mw = 0$ by definition (55). To demonstrate (56) we apply the definition (55) 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 20.

The last three statements of the proposition are direct consequences of (56). 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 (56) 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 (55) 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 (55) as the Euclidean projection of $x + \delta u$.

It is interesting to note that it is not necessary to have $\delta \leq \bar{\delta}$ for Proposition 22 to be applicable.

4.2.2 Algorithm HCE

The formal statement of algorithm HCE is as follows:

Algorithm HCE

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

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

$$I(\delta) \triangleq \left\lceil \frac{9}{2\beta_C^2\delta^2} \ln\left(\frac{1}{2\tau_C\delta^2}\left(1 + \frac{1}{\beta_C\delta}\right)\right) \right\rceil. \tag{57}$$

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

$$\left\lceil \log_2 \left(\frac{\|M\|}{\rho(M)} \right) \right\rceil + 2 \tag{58}$$

iterations, performing at most

$$\frac{4}{3} \left\lceil \frac{216\|M\|^2}{\rho(M)^2 \beta_C^2} \ln \left(\frac{40\|M\|}{\rho(M)\tau_C \beta_C} \right) \right\rceil + \left\lceil \log_2 \left(\frac{\|M\|}{\rho(M)} \right) \right\rceil + 2 \tag{59}$$

iterations of algorithm GVNA.

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

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

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 $v = -\delta M u - Mx = -M(x + \delta u)$. From Proposition 22, 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

$$\epsilon = \frac{1}{2}\delta \tau_C \frac{\rho(M)^2}{\|M\|}. \tag{60}$$

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 (60).

Suppose $0 < \delta \leq \frac{1}{2}\frac{\rho(M)}{\|M\|}$. Let $\bar{\delta}$ be as defined in (53). Applying Lemma 21 for $z = -Mu$ we conclude that the system (HCE$_\delta$) is feasible for any $\delta \in [0, \bar{\delta}]$, and $\bar{\delta} \geq \frac{\rho(M)}{\|Mu\|} \geq \frac{\rho(M)}{\|M\|} > \delta$. Hence the system (HCE$_\delta$) is feasible, and furthermore

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

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

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

Since the system (HCE_δ) is feasible, from Proposition 11 it must be true that algorithm GVNA with STOP = STOP3 will perform $I = I(\delta)$ iterations, where

$$I(\delta) \triangleq \left\lceil \frac{9}{2\beta_C^2\delta^2} \ln\left(\frac{1}{2\tau_C\delta^2}\left(1+\frac{1}{\beta_C\delta}\right)\right)\right\rceil \geq \frac{18\|M\|^2}{\rho(M)^2\beta_C^2} \ln\left(\frac{2\|M\|^2}{\rho(M)^2\tau_C}\left(1+\frac{1}{\beta_C\delta}\right)\right), \quad (61)$$

since $\delta \leq \frac{1}{2}\frac{\rho(M)}{\|M\|}$. Applying Lemma 15 we conclude that after $I(\delta)$ iterations of GVNA the residual $v^{I(\delta)}$ satisfies:

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

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

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

$$K = \left\lceil \log_2\left(\frac{\|M\|}{\rho(M)}\right)\right\rceil + 2 \quad (62)$$

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

$$\delta^k > \frac{1}{4}\frac{\rho(M)}{\|M\|}. \quad (63)$$

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

$$\sum_{k=1}^{K} I(\delta^k) = \sum_{k=1}^{K}\left\lceil \frac{9\cdot 4^k}{8\beta_C^2}\ln\left(\frac{4^k}{8\tau_C}\left(1+\frac{2^{k-1}}{\beta_C}\right)\right)\right\rceil. \quad (64)$$

It can be shown by analyzing the geometric series $\sum_{k=1}^{K} 4^k$ that the sum in (64) satisfies $\sum_{k=1}^{K} I(\delta^k) \le \frac{4}{3} I(\delta^K) + K$. Therefore

$$\begin{aligned}
\sum_{k=1}^{K} I(\delta^k) &\le \frac{4}{3}\left\lceil \frac{9}{2\beta_C^2(\delta^K)^2} \ln\left(\frac{1}{2\tau_C(\delta^K)^2}\left(1+\frac{1}{\beta_C\delta^K}\right)\right)\right\rceil + K \\
&\le \frac{4}{3}\left\lceil \frac{72\|M\|^2}{\rho(M)^2\beta_C^2} \ln\left(\frac{8\|M\|^2}{\rho(M)^2\tau_C}\left(1+\frac{4\|M\|}{\rho(M)\beta_C}\right)\right)\right\rceil + \left\lceil \log_2\left(\frac{\|M\|}{\rho(M)}\right)\right\rceil + 2 \\
&\le \frac{4}{3}\left\lceil \frac{72\|M\|^2}{\rho(M)^2\beta_C^2} \ln\left(\frac{40\|M\|^3}{\rho(M)^3\tau_C\beta_C}\right)\right\rceil + \left\lceil \log_2\left(\frac{\|M\|}{\rho(M)}\right)\right\rceil + 2 \\
&\le \frac{4}{3}\left\lceil \frac{216\|M\|^2}{\rho(M)^2\beta_C^2} \ln\left(\frac{40\|M\|}{\rho(M)\tau_C\beta_C}\right)\right\rceil + \left\lceil \log_2\left(\frac{\|M\|}{\rho(M)}\right)\right\rceil + 2. \qquad (65)
\end{aligned}$$

The first inequality in (65) follows from (63). 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 22 combined with (63) 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

$$\begin{aligned}
\frac{\|w\|}{\operatorname{dist}(w,\partial C)} &\le \frac{\frac{1}{2}\delta^K\tau_C + \frac{1}{\beta_C} + \delta^K}{\frac{1}{2}\tau_C\delta^K} = 2\left(\frac{1}{2} + \frac{1}{\beta_C\tau_C\delta^K} + \frac{1}{\tau_C}\right) \\
&\le 2\left(\frac{1}{2} + \frac{4\|M\|}{\rho(M)\beta_C\tau_C} + \frac{1}{\tau_C}\right) \le \frac{11\|M\|}{\rho(M)\beta_C\tau_C},
\end{aligned}$$

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

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

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

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

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 for the cone C_X.

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):

$$\begin{array}{l} -b\theta + Ax = 0 \\ \theta \geq 0, \ x \in C_X. \end{array} \tag{66}$$

System (66) is of the form (HCE) of (48) 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 24 *Suppose (FP_d) of (1) is feasible and $\rho(d) > 0$. Then the system (66) is feasible, M has full rank, and we have*

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

where $\rho(M)$ is defined in (49).

Proof: Feasibility of the system (66) 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 [29]. Since $\rho(M) = \rho(d) > 0$, Remark 20 implies that M has full rank. ∎

We use algorithm HCE to find a sufficiently interior solution of the system (66) 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 (66). 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 25 *Suppose (FP_d) is feasible and $\rho(d) > 0$. Then algorithm FCLS will terminate in at most*

$$\frac{4}{3}\left\lceil \frac{216\mathcal{C}(d)^2}{\beta^2} \ln\left(\frac{80\mathcal{C}(d)}{\tau\beta}\right)\right\rceil + \lceil \log_2 \mathcal{C}(d) \rceil + 2 \tag{67}$$

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 $\alpha \triangleq \text{dist}(\tilde{w}, \partial C) = \text{dist}\left((\tilde{\theta}, \tilde{x}), \partial(\Re_+ \times C_X)\right)$.

From Theorem 23 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 23 we conclude that the vector $\tilde{w} = (\tilde{\theta}, \tilde{x})$ returned by algorithm HCE in Step 1 satisfies:

$$-b\tilde{\theta} + A\tilde{x} = 0, \ (\tilde{\theta}, \tilde{x}) \in \Re_+ \times C_X, \ \alpha \geq \frac{\tau_C \rho(M)}{8\|M\|} \geq \frac{\tau}{16\mathcal{C}(d)}, \tag{68}$$

$$|\tilde{\theta}| + \|\tilde{x}\| \leq \frac{5}{2\beta_C} = \frac{5}{2\beta}, \quad \frac{\|(\tilde{\theta}, \tilde{x})\|}{\alpha} \leq \frac{11\|M\|}{\rho(M)\beta_C\tau_C} \leq \frac{22\mathcal{C}(d)}{\beta\tau}. \tag{69}$$

Note in particular that (68) 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 $r \triangleq \frac{\alpha}{\|\tilde{w}\|}(1 + \|\hat{x}\|)$. Then a simple application of (69) implies that $r \geq \frac{\beta\tau}{22C(d)}$. Further, let $p \in X$ be an arbitrary vector satisfying $\|p\| \leq r$. Then

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

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, $\mathrm{dist}(\hat{x}, \partial C_X) \geq r \geq \frac{\beta\tau}{22C(d)}$, establishing *3*.

Finally,

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

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):

$$\begin{array}{l} -br + Ax = 0 \\ r + \bar{f}^t x = 1, \\ r \geq 0, \;\; x \in C_X. \end{array} \tag{70}$$

The alternative system to (70) is given by

$$\begin{array}{ll} -b^t s & > 0 \\ A^t s & \in \mathrm{int} C_X^*. \end{array} \tag{71}$$

System (71) 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:

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

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

Proposition 26 *Suppose the system (FP_d) is infeasible and $\rho(d) > 0$. Then the system (70) 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 (44).

Proof: Infeasibility of the system (70) follows from Proposition 3. 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 (70) is infeasible $r(M,0)$ is computed using (33) as

$$r(M,0) = \begin{array}{ll} \min & \|0 - M(r,x)\| \\ & r + \bar{f}^t x = 1 \\ & r \geq 0,\ x \in C_X \end{array} = \begin{array}{ll} \min & \|br - Ax\| \\ & r + \bar{f}^t x = 1 \\ & r \geq 0,\ x \in C_X, \end{array} \tag{72}$$

which is exactly program $P_g(d)$ of [12] (for the case when $C_Y = \{0\}$). Therefore, applying Theorem 3.9 of [12] 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 (71) 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 (71). The algorithm will return a vector s.

  Step2 Return s (a reliable solution of (SA_d)).

Lemma 27 *Suppose (FP_d) is infeasible and $\rho(d) > 0$. Then algorithm ICLS will terminate in at most*

$$\left\lfloor \frac{16\mathcal{C}(d)^2}{\beta^2} \right\rfloor \tag{73}$$

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

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

Proof: From Theorem 19 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

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

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 is designed to run both algorithms FCLS and ICLS in parallel, and will terminate 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 25 and 27 we obtain the following complexity analysis of algorithm CLS:

Theorem 28 *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:

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

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

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

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:

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

Proof: The proof is an immediate consequence of Lemmas 25 and 27. 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 $g = -\delta M u$ 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 28, is the total number of iterations of algorithm GVNA called in algorithm CLS. In this perspective, algorithm CLS is "elementary" in that the mathematics of 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, see Remark 29.

Remark 29 *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 ($CSOP_{C_X}$). 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 Euclidean projections using formula (55). This computation cannot be considered elementary since, in particular, it involves computing an inverse of a square matrix MM^t which 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 $P \triangleq I - M^t(MM^t)^{-1}M$ "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 23), the multiplications involving matrix P will not increase the computation time significantly even though matrix P is not likely to have a nice sparsity structure.

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 [8]. 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 [25] and can be interpreted as introducing scaled slack variables for the linear constraints. This approach is extended to problems in standard dual form in [8]. 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 [8].

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 28 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] S. Agmon. The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6:382–392, 1954.

[2] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.

[3] Daniel Bienstock. Experiments with a network design algorithm using ϵ- approximate linear programs. December 1996.

[4] C. Chen and O.L. Mangasarian. Smoothing methods for convex inequalities and linear complementarity problems. *Mathematical Programming*, 71(1):51–70, 1995.

[5] G.B. Dantzig. Converting a converging algorithm into a polynomially bounded algorithm. Technical Report SOL 91-5, Stanford University, 1991.

[6] G.B. Dantzig. 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, 1992.

[7] B.C. Eaves. Piecewise linear retractions by reflection. *Linear Algebra and its Applications*, 7:93–98, 1973.

[8] Marina Epelman. *Complexity, Condition Numbers, and Conic Linear Systems.* PhD thesis, Massachusetts Institute of Technology, 1999. Forthcoming.

[9] Marina Epelman and Robert M. Freund. 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, February 1997.

[10] Sharon Filipowski. 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, May 1994.

[11] Sharon Filipowski. 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, December 1994.

[12] Robert M. Freund and Jorge R. Vera. Some characterizations and properties of the "distance to ill-posedness" in conic linear systems. Technical Report W.P. #3862-95-MSA, Sloan School of Management, Massachusetts Institute of Technology, 1995.

[13] Robert M. Freund and Jorge R. Vera. Condition-based complexity of convex optimization in conic linear form via the ellipsoid algorithm. *SIAM Journal on Optimization* (to appear), 1998.

[14] J.L. Goffin. The relaxation method for solving systems of linear inequalities. *Mathematics of Operations Research*, 5(3):388–414, 1980.

[15] Andrew V. Goldberg, Jeffrey D. Oldham, Serge Plotkin, and Cliff Stein. An implementation of a combinatorial approximation algorithm for minimum-cost multicommodity flow, December 1997.

[16] Michael D. Grigoriadis and Leonid G. Khachiyan. Fast approximation schemes for convex programs with many blocks and coupling constraints. *SIAM Journal on Optimization*, 4(1):86–107, 1994.

[17] Michael D. Grigoriadis and Leonid G. Khachiyan. Coordination complexity of parallel price-directive decomposition. *Mathematics of Operations Research*, 21(2):321–340, 1996.

[18] Richard A. Horn and Charles R. Johnson. *Matrix Analysis.* Cambridge University Press, New York, 1985.

[19] David Karger and Serge Plotkin. 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*, pages 18–25, 1995.

[20] T.S. Motzkin and I.J. Schoenberg. The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6:393–404, 1954.

[21] Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1994.

[22] Manuel A. Nunez and Robert M. Freund. Condition measures and properties of the central trajectory of a linear program. Technical Report W.P. #3889-96-MSA, Sloan School of Management, Massachusetts Institute of Technology, 1996.

[23] Javier Peña. Understanding the geometry of infeasible perturbations of a conic linear system. Preprint.

[24] Javier Peña. Computing the distance to infeasibility: theoretical and practical issues. Technical report, Cornell University, October 1997.

[25] Javier Peña and James Renegar. Quickly computing backward-approximate solutions for ill-conditioned systems of linear inequalities and forward-approximate solutions for well-conditioned systems, May 1998.

[26] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems, February 1995.

[27] James Renegar. Some perturbation theory for linear programming. *Mathematical Programming*, 65(1):73–91, 1994.

[28] James Renegar. Incorporating condition measures into the complexity theory of linear programming. *SIAM Journal on Optimization*, 5(3):506–524, 1995.

[29] James Renegar. Linear programming, complexity theory, and elementary functional analysis. *Mathematical Programming*, 70(3):279–351, 1995.

[30] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

[31] F. Rosenblatt. On the convergence of reinforcement procedures in simple perceptrons. Report, VG-1196-G-4, Cornell Aeronautical Laboratory, Buffalo, NY, 1960.

[32] F. Rosenblatt. Perceptron simulation experiments. *Proceedings of the Institute of Radio Engineers*, 48:301–309, 1960.

[33] F. Rosenblatt. *Principles of Neurodynamics.* Spartan Books, Washington, DC, 1962.

[34] Jorge R. Vera. Ill-posedness and the computation of solutions to linear programs with approximate data. Technical Report, Cornell University, May 1992.

[35] Jorge R. Vera. *Ill-Posedness in Mathematical Programming and Problem Solving with Approximate Data.* PhD thesis, Cornell University, 1992.

[36] Jorge R. Vera. Ill-posedness and the complexity of deciding existence of solutions to linear programs. *SIAM Journal on Optimization*, 6(3):549–569, 1996.

[37] R. Vera, Jorge. On the complexity of linear programming under finite precision arithmetic. *Mathematical Programming*, 80(1):91–123, 1998.

[38] Jorge Villavicencio and Michael D. Grigoriadis. Approximate Lagrangean decomposition with a modified Karmarkar logarithmic potential. Technical Report LCsR-TR 258, Rutgers University, June 1996.