An Efficient Rescaled Perceptron Algorithm for Conic Systems

An Efficient Re-scaled Perceptron Algorithm for Conic Systems

Alexandre Belloni
Fuqua School of Business, Duke University, One Towerview Drive, Box 90120, Durham, NC 27708-0120
email: abn5@duke.edu http://www.duke.edu/~abn5/

Robert M. Freund
MIT Sloan School of Management, E53-357, 50 Memorial Drive, Cambridge, Massachusetts 02142
email: rfreund@mit.edu http://web.mit.edu/rfreund/www

Santosh Vempala
College of Computing, Klaus 2222, Georgia Tech, 266 Ferst Drive, Atlanta, Georgia 30332
email: vempala@cc.gatech.edu http://www.cc.gatech.edu/~vempala/

The classical perceptron algorithm is an elementary row-action/relaxation algorithm for solving a homogeneous linear inequality system $Ax > 0$. A natural condition measure associated with this algorithm is the Euclidean width τ of the cone of feasible solutions, and the iteration complexity of the perceptron algorithm is bounded by $1/\tau^2$, see Rosenblatt 1962 [20]. Dunagan and Vempala [5] have developed a re-scaled version of the perceptron algorithm with an improved complexity of $O(n \ln(1/\tau))$ iterations (with high probability), which is theoretically efficient in τ, and in particular is polynomial-time in the bit-length model. We explore extensions of the concepts of these perceptron methods to the general homogeneous conic system $Ax \in \textbf{int } K$ where K is a regular convex cone. We provide a conic extension of the re-scaled perceptron algorithm based on the notion of a *deep-separation oracle* of a cone, which essentially computes a certificate of strong separation. We show that the re-scaled perceptron algorithm is theoretically efficient if an efficient deep-separation oracle is available for the feasible region. Furthermore, when K is the cross-product of basic cones that are either half-spaces or second-order cones, then a deep-separation oracle is available and hence the re-scaled perceptron algorithm is theoretically efficient. When the basic cones of K include semi-definite cones, then a probabilistic deep-separation oracle for K can be constructed that also yields a theoretically efficient version of the re-scaled perceptron algorithm.

Key words: Convex Cones ; Perception ; Conic System ; Separation Oracle

MSC2000 Subject Classification: Primary: 52A20 , 90C60 ; Secondary: 90C25

OR/MS subject classification: Primary: Convexity ; Secondary: Random Walk

1. Introduction. We consider the problem of computing a solution of the following conic system

$$\left\{ \begin{array}{rcl} Ax & \in & \textbf{int } K \\ x & \in & X \end{array} \right. \tag{1}$$

where X and Y are n- and m-dimensional linear subspaces, respectively, $A : X \to Y$ is a linear operator and $K \subset Y$ is a regular closed convex cone. We refer to this problem as the "conic inclusion" problem, we call K the *inclusion cone* and we call $\mathcal{F} := \{x \in X : Ax \in K\}$ the *feasibility cone*. The goal is to compute an interior element of the feasibility cone $\mathcal{F}$. Important special cases of this format include feasibility problem instances for linear programming (LP), second-order cone programming (SOCP) and positive semi-definite programming (SDP).

The ellipsoid method ([12]), the random walk method ([3]), and interior-point methods (IPMs) ([11], [14]) are examples of methods which solve (1) in polynomial-time. Nonetheless, these methods differ substantially in their representation requirement as well as in their practical performance. For example, a membership oracle suffices for the ellipsoid method and the random walk method, while a special barrier function for K is required to implement an IPM. The latter is by far the most successful algorithm for conic programming in practice: for example, applications of SDP range over several fields including optimal control, eigenvalue optimization, combinatorial optimization and many others, see [22].

In the case when $X = \mathbb{R}^n$ and $K = \mathbb{R}^m_+$, we recover the original setting of a homogeneous system of linear inequalities. Within this context, another alternative method is the perceptron algorithm [20]. It is well-known that this simple method terminates after a finite number of iterations which can be bounded by the square of the inverse of the *width* τ of the feasibility cone $\mathcal{F}$. Although occasionally attractive from a practical point of view due to its simplicity, the perceptron algorithm is not considered theoretically efficient since the width τ can be exponentially small in the size of the instance in the bit-length model. Dunagan and Vempala ([5]) combined the perceptron algorithm with a sequence of re-scalings constructed from near-feasible solutions. These re-scalings gradually increase τ on average and the resulting re-scaled perceptron algorithm has complexity $O(n \ln(1/\tau))$ iterations (with high probability), which is theoretically efficient.

Herein we extend the re-scaled perceptron algorithm proposed in [5] to the conic setting of (1). Although the probabilistic analysis is similar, this is not the case for the remainder of the analysis. In particular, we show that the improvement obtained in [5] arises from the use of a *deep-separation oracle*, which is stronger than the usual separation oracle used in the classical perceptron algorithm. In the case of a system of linear inequalities studied in [5], there is no difference between the implementation of both oracles. However, this difference is significant for more general cones.

We investigate in detail ways to construct a deep-separation oracle for several classes of cones, since it is the driving force of the re-scaled perceptron algorithm. We establish important properties of the deep-separation oracle and its implementation for several classes (including the case when K is the cross-product of half-spaces and second-order cones). When the basic cones comprising K include semi-definite cones, we present a probabilistic version of a deep-separation oracle that also yields a theoretically efficient version of the re-scaled perceptron algorithm.

We start in Section 2 with properties of convex cones, oracles, and the definition of a deep-separation oracle. Section 3 generalizes the classical perceptron algorithm to the conic setting, and Section 4 extends the re-scaled perceptron algorithm of [5] to the conic setting. Section 5 contains the probabilistic and complexity analysis of the re-scaled perceptron algorithm, which reviews some material from [5] for completeness. Section 6 is devoted to methods for constructing a deep-separation oracle for both specific and general cones.

The perceptron algorithm is a greedy procedure that updates the current proposed solution by using any violated inequality. The number of iterations is finite but can be exponential. The modified perceptron algorithm (proposed in [4], used in [5]) is a similar updating procedure that only uses inequalities that are violated by at least some fixed threshold. Although this procedure is not guaranteed to find a feasible solution, it finds a near-feasible solution with the guarantee that no constraint is violated by more than the threshold and the number of steps to convergence is proportional to the inverse square of the threshold, independent of the conditioning of the initial system. The key idea in [5] is that such a near-feasible solution can be used to improve the width of the original system by a multiplicative factor. As we show in this paper, this analysis extends naturally to the full generality of conic systems.

The main difficulty is in identifying a constraint that is violated by more than a fixed threshold by the current proposed solution, precisely what we call a deep-separation oracle. This is not an issue in the linear setting (one simply checks each constraint). For conic systems, the deep-separation itself is a conic feasibility problem! It has the form: find $w \in K^*$, the dual of the original inclusion cone, such that w satisfies a single second-order conic constraint. Our idea is to apply the re-scaled perceptron algorithm to this system which is considerably simpler than $\mathcal{F}$. What we can prove is that when K is composed of basic cones that are either half-spaces or second-order cones, such a deep-separation oracle is readily available. When the basic cones comprising K include semi-definite cones, we show that there is a probabilistic version of a deep-separation oracle. This probabilistic deep-separation oracle still yields a theoretically efficient version of the re-scaled perceptron algorithm.

2. Preliminaries

2.1 Notation For simplicity we confine our analysis to finite dimensional linear spaces. Let X and Y denote linear spaces with finite dimension n and m, respectively, with inner product operators denoted generically by $\langle \cdot, \cdot \rangle$. All norms are induced by inner-products: $\|v\| := \sqrt{\langle v, v \rangle}$. For $\bar{x} \in X$, $B(\bar{x}, r)$ will denote the ball centered at $\bar{x}$ with radius r, and analogously for Y. Let **cl** S and **int** S denote the closure and interior of a set S, respectively. Let $A : X \to Y$ denote a linear operator, and $A^* : Y \to X$ denote the adjoint operator associated with A.

2.2 Convex Cones Let C be a convex cone. The dual cone of C is defined as

$$C^* = \{d : \langle x, d \rangle \geq 0, \text{ for all } x \in C\} \tag{2}$$

and $\mathbf{ext}C$ denotes the set of extreme rays of C. A cone is pointed if it contains no lines. We say that C is a *regular* cone if C is a pointed closed convex cone with non-empty interior. It is elementary to show that

C is regular if and only if C^* is regular. Given a regular convex cone C, we use the following geometric (condition) measure:

Definition 2.1 *If C is a regular cone in X, the* width *of C is given by*

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

Furthermore the center *of C is any vector $\bar{z}$ that attains the above maximum, normalized so that $\|\bar{z}\| = 1$.*

We will be particularly interested in the following three classes of cones: the non-negative orthant $\mathbb{R}^k_+ := \{x \in \mathbb{R}^k : x \geq 0\}$, the second order cone denoted by $Q^k := \{x \in \mathbb{R}^k : \|(x_1, x_2, \ldots, x_{k-1})\| \leq x_k\}$, and the cone of positive semi-definite matrices $S^{k\times k}_+ := \{X \in S^{k\times k} : \langle v, Xv\rangle \geq 0 \text{ for all } v \in \mathbb{R}^k\}$ where $S^{k\times k} := \{X \in \mathbb{R}^{k\times k} : X = X^T\}$. Both $\mathbb{R}^k_+$ and Q^k are defined on $\mathbb{R}^k$ whose inner product is the usual scalar product $\langle v, w\rangle = v^T w = \sum_{i=1}^m v_i w_i$. From this inner product it is straightforward to show that both $\mathbb{R}^k_+$ and Q^k are self dual, and that their widths are $1/\sqrt{k}$ and $1/\sqrt{2}$, respectively. The semidefinite cone $S^{k\times k}_+$ is defined on the linear space of symmetric matrices $S^{k\times k}$. For this space, it will be convenient to depart from our standard notation and represent points in $S^{k\times k}$ using capital letters such as X, for example. For $S^{k\times k}$ we assign the trace inner product $\langle W, V\rangle = \textbf{Trace}\ (W^T V) = \sum_{i,j\in\{1,\ldots,k\}} W_{ij}V_{ij}$, yielding the Frobenius norm as the inner-product norm $\|X\| = \sqrt{\langle X, X\rangle} = \sqrt{\sum_{i,j\in\{1,\ldots,k\}} X_{ij}^2}$. (The trace inner product and consequential Frobenius norm are standard in the modern treatment of semidefinite optimization, see [22].) Using the trace inner product one easily establishes that $S^{k\times k}_+$ is self-dual and the width of $S^{k\times k}_+$ is $1/\sqrt{k}$. We also define the Löwner partial ordering "$\succeq$" on $S^{k\times k}$ as $X \succeq W$ if and only if the matrix $X - W \in S^{k\times k}_+$.

The following characterization will be used in our analysis.

Lemma 2.1 *Suppose that $C \subset Y$ is a closed convex cone and $M \in L(X, Y)$. Let $\mathcal{G} = \{x : Mx \in C\}$ and let $T = \{M^*\lambda : \lambda \in C^*\}$. Then* **cl** $(T) = \mathcal{G}^*$.

Lemma 2.1 is a special case of a more general result about dual cones involving linear operators, see Theorem 3.1 of Berman [2]. The following proof of Lemma 2.1 is included for completeness.

Proof. ($\subseteq$) Let $\lambda \in C^*$. Then for every x satisfying $Mx \in C$, $\langle x, M^*\lambda\rangle = \langle Mx, \lambda\rangle \geq 0$, since $Mx \in C$ and $\lambda \in C^*$. Thus, **cl** $(T) \subseteq \mathcal{G}^*$ since $\mathcal{G}^*$ is closed.

($\supseteq$) First note that **cl** (T) is a nonempty closed convex set. Assume that there exists $y \in \mathcal{G}^* \backslash$**cl** (T). Thus there exists $h \neq 0$ satisfying $\langle h, y\rangle < 0$ and $\langle h, w\rangle \geq 0$ for all $w \in$ **cl** (T). Notice that $\langle h, M^*\lambda\rangle \geq 0$ for all $\lambda \in C^*$, which implies that $Mh \in C$ and so $h \in \mathcal{G}$. On the other hand, since $y \in \mathcal{G}^*$, it follows that $\langle h, y\rangle \geq 0$, contradicting $\langle h, y\rangle < 0$. □

The question of sets of the form T being closed has been recently studied by Pataki [15]. Necessary and sufficient conditions for T to be a closed set are given in [15] when C^* belongs to a class called "nice cones," a class which includes polyhedra and self-scaled cones. Nonetheless, the set T may fail to be closed even in simple cases, as the following example shows.

Example 2.1 *Let $C^* = Q^3 = \{(\lambda_1, \lambda_2, \lambda_3) \mid \|(\lambda_1, \lambda_2)\| \leq \lambda_3\}$ and $M = \begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$. In this case, $T = \{M^*\lambda \mid \lambda \in C^*\} = \{(-\lambda_1 + \lambda_3, \lambda_2) \mid \|(\lambda_1, \lambda_2)\| \leq \lambda_3\}$. It is easy to verify that $(0, 1) \notin T$ but $(\varepsilon, 1) \in T$ for every $\varepsilon > 0$ (set $\lambda_1 = \frac{1}{2\varepsilon} - \frac{\varepsilon}{2}$, $\lambda_2 = 1$, and $\lambda_3 = \frac{1}{2\varepsilon} + \frac{\varepsilon}{2}$), which shows that T is not closed.*

The following property of convex cones is well-known, but is presented and proved herein both for completeness as well as for conformity to our notation.

Lemma 2.2 *Suppose that $C \subset Y$ is a closed convex cone. $B(z, r) \subseteq C$ if and only if $\langle d, z\rangle \geq r\|d\|$ for all $d \in C^*$.*

Proof. Suppose $B(z,r) \subset C$. Let $d \in C^*$, $d \neq 0$. Then, $z - r\frac{d}{\|d\|} \in C$ and since $d \in C^*$, $\left\langle d, z - r\frac{d}{\|d\|} \right\rangle \geq 0$. Thus, $\langle d, z\rangle \geq r\frac{\langle d,d\rangle}{\|d\|} = r\|d\|$. Conversely, suppose $\langle d, z\rangle \geq r\|d\|$ for every $d \in C^*$. Let v satisfy $\|v\| \leq r$. Assume $z + v \notin C$, then there exists $d \in C^*$, $\langle d, z+v\rangle < 0$. Therefore $\langle d, z\rangle < -\langle d, v\rangle \leq r\|d\|$, which contradicts $\langle d, z\rangle \geq r\|d\|$. □

Lemma 2.3 *Let $B : X \to X$ be a self-adjoint invertible linear operator. Let $\mathcal{F}_A = \{x \in X : Ax \in K\}$ and $\mathcal{F}_{AB} = \{x \in X : ABx \in K\}$. Then $\mathcal{F}^*_{AB} = B^*\mathcal{F}^*_A$.*

Proof. From Lemma 2.1 we have $\mathcal{F}^*_{AB} = \mathbf{cl}\left(\{B^*A^*\lambda : \lambda \in K^*\}\right) = \mathbf{cl}\left(B^*\{A^*\lambda : \lambda \in K^*\}\right)$. Therefore we need to prove that

$$\mathbf{cl}\left(B^*\{A^*\lambda : \lambda \in K^*\}\right) = B^*\mathbf{cl}\left(\{A^*\lambda : \lambda \in K^*\}\right)$$

which follows since B^* is invertible and Theorem 9.1 from [19]. □

2.3 Oracles In our algorithms and analysis we will distinguish two different types of oracles.

Definition 2.2 *An* interior separation oracle *for a convex set $S \subset \mathbb{R}^n$ is a subroutine that given a point $x \in \mathbb{R}^n$, identifies if $x \in \mathbf{int}\ S$ or returns a vector $d \in \mathbb{R}^n$, $d \neq 0$, such that*

$$\langle d, x\rangle \leq \langle d, y\rangle \ \text{for all } y \in S\ .$$

Definition 2.3 *For a fixed positive scalar t, a* deep-separation oracle *for a cone $C \subset \mathbb{R}^n$ is a subroutine that given a non-zero point $x \in \mathbb{R}^n$, either*

(I) correctly identifies that $\dfrac{\langle d, x\rangle}{\|d\|\|x\|} \geq -t$ *for all* $d \in \mathbf{ext}C^*$

or

(II) returns a vector $d \in C^*$, $d \neq 0$, satisfying $\dfrac{\langle d, x\rangle}{\|d\|\|x\|} \leq -t.$

Definition 2.2 is standard in the literature, whereas Definition 2.3 is new as far as we know. Our motivation for this definition arises from a relaxation of the orthogonality characterization of a convex cone. For $d, x \neq 0$ let $\cos(d,x)$ denote the cosine of the angle between d and x, i.e., $\cos(d,x) = \frac{\langle d,x\rangle}{\|d\|\|x\|}$. Notice that $x \in C$ if and only if $\cos(d,x) \geq 0$ for all $d \in C^*$ if and only if $\cos(d,x) \geq 0$ for all $d \in \mathbf{ext}C^*$. The latter characterization states that $\frac{\langle d,x\rangle}{\|d\|\|x\|} \geq 0$ for all $d \in \mathbf{ext}C^*$. Condition (I) of the deep-separation oracle relaxes the cosine condition from 0 to $-t$. The following example illustrates that the perceptron improvement algorithm described in [5] corresponds to a deep-separation oracle for a linear inequality system.

Example 2.2 *Let $C = \{x \in \mathbb{R}^n : Mx \geq 0\}$ where M is an $m \times n$ matrix none of whose rows are zero. Notice that $C^* = \{M^*\lambda : \lambda \geq 0\}$ is the conic hull of the rows of M, and the extreme rays of C^* are a subset of the rows of M. Therefore a deep-separation oracle for C can be constructed by identifying for a given $x \neq 0$ if there is an index $i \in \{1, \ldots, m\}$ for which $\frac{\langle M_i, x\rangle}{\|M_i\|\|x\|} \leq -t$ and returning $M_i/\|M_i\|$ in such a case. Notice that we do not need to know which vectors M_i are extreme rays of C^*; if m is not excessively large it is sufficient to simply check the aforementioned inequality for every row index i.*

Remark 2.1 *It might seem odd that condition (I) involves "only" the extreme rays of C^*. However, in many particular conic structures arising in practice, a super-set of the extreme rays of the dual cone C^* is at least partially accessible, as is the case when $C = \{x : Mx \geq 0\}$ where this super-set is comprised of the row vectors of M. Indeed, suppose we replace condition (I) by the seemingly more convenient condition "$\frac{\langle d,x\rangle}{\|d\|\|x\|} \geq -t$ for all $d \in C^*$." Utilizing Lemma 2.1, this condition is met by checking $-t \leq \min_\lambda\{\langle M^*\lambda, x/\|x\|\rangle : \|M^*\lambda\| \leq 1,\ \lambda \geq 0\}$, and taking a dual yields $-t \leq \max_w\{-\|w - x/\|x\|\| : Mw \geq 0\}$. We see that this latter optimization problem simply tests if $x/\|x\|$ is at most distance t from the cone C, which itself is at least as hard as computing a non-trivial point in C.*

Remark 2.2 *It turns out that conditions (I) and (II) might each be strictly satisfiable. Let* $C = \{x : Mx \geq 0\}$ *where* $M = \begin{bmatrix} -2 & 3 \\ 3 & -2 \\ 0 & 1 \end{bmatrix}$. *Then C has an interior solution, and let $t = 3/4$. It is straightforward to check that $x = (-1,-1)$ satisfies $\frac{\langle M_i, x\rangle}{\|M_i\|\|x\|} > -t$ for every i, whereby condition (I) is satisfied strictly. Furthermore, $\bar{d} = (1,1) \in C^*$ and satisfies $\frac{\langle \bar{d}, x\rangle}{\|\bar{d}\|\|x\|} < -t$, thus showing that condition (II) is also satisfied strictly. Of course $\bar{d} \notin$ **ext**C^*, thus highlighting the importance of the role of extreme rays.*

3. Perceptron Algorithm for a Conic System

The classical perceptron algorithm was proposed to solve a homogeneous system of linear inequalities (1) with $K = \mathbb{R}^m_+$. It is well-known that the algorithm has finite termination in at most $\lfloor 1/\tau_{\mathcal{F}}^2 \rfloor$ iterations, see Rosenblatt 1962 [20]. This complexity bound can be exponential in the bit-model.

Our starting point herein is to show that the classical perceptron algorithm can be easily extended to the case of a conic system of the form (1).

Perceptron Algorithm for a Conic System
(a) Let x be the origin in X. Repeat:
(b) If $Ax \in$ **int** K, Stop. Otherwise, call interior separation oracle for $\mathcal{F}$ at x, returning $d \in \mathcal{F}^*$, $d \neq 0$, such that $\langle d, x\rangle \leq 0$, and set $x \leftarrow x + d/\|d\|$.

This algorithm presupposes the availability of an interior separation oracle for the feasibility cone $\mathcal{F}$. In the typical case when the inclusion cone K has an interior separation oracle, this oracle can be used to construct an interior separation oracle for $\mathcal{F}$: if $x \notin$ **int** $\mathcal{F}$, then $Ax \notin$ **int** K and there exists $\lambda \in K^*$, $\lambda \neq 0$, satisfying $\langle \lambda, Ax\rangle \leq 0$, whereby $d = A^*\lambda$ satisfies $\langle d, x\rangle \leq 0$, $d \in \mathcal{F}^*$, and $d \neq 0$. (If $d = 0$, it is straightforward to show that **int** $\mathcal{F} = \emptyset$.)

Exactly as in the case of linear inequalities, we have the following iteration bound for this algorithm.

Lemma 3.1 *The perceptron algorithm for a conic system will compute a solution of (1) in at most $\lfloor 1/\tau_{\mathcal{F}}^2 \rfloor$ iterations.*

Proof. Consider the potential function $\pi(x) = \langle x, \bar{z}\rangle / \|x\|$, and note that $\pi(x) \leq 1$ for all $x \neq 0$, where $\tau_{\mathcal{F}}$ is the width of the feasibility cone $\mathcal{F}$ and $\bar{z}$ is the center of $\mathcal{F}$. If the algorithm does not stop at **(b)**, we update x to $x + \bar{d}$ where $\bar{d} := d/\|d\|$, whereby

$$\left\langle x + \bar{d}, \bar{z}\right\rangle = \langle x, \bar{z}\rangle + \left\langle \bar{d}, \bar{z}\right\rangle \geq \langle x, \bar{z}\rangle + \tau_{\mathcal{F}},$$

and

$$\|x + \bar{d}\|^2 = \langle x, x\rangle + 2\left\langle x, \bar{d}\right\rangle + \left\langle \bar{d}, \bar{d}\right\rangle \leq \langle x, x\rangle + 1,$$

since $\left\langle x, \bar{d}\right\rangle \leq 0$, $\left\langle \bar{d}, \bar{d}\right\rangle = 1$, and $\left\langle \bar{d}, \bar{z}\right\rangle \geq \tau_{\mathcal{F}}$ from Lemma 2.2.

After k iterations, the potential function is at least $k\tau_{\mathcal{F}}/\sqrt{k}$. After more than $\lfloor 1/\tau_{\mathcal{F}}^2 \rfloor$ iterations, the potential function would be greater than one, a contradiction. Thus, the algorithm must terminate after at most $\lfloor 1/\tau_{\mathcal{F}}^2 \rfloor$ iterations, having computed a solution of (1). □

Example 3.1 *Consider the semidefinite cone $K = S^{k\times k}_+$ and the linear operator $A : \mathbb{R}^n \to S^{k\times k}$. Suppose that $Ax \notin$ **int** K. In order to compute a direction $d \in \mathcal{F}^*$, we start by computing any eigenvector v of the symmetric matrix Ax associated with a non-positive eigenvalue. Then the vector $d = A^*(vv^T)$ will satisfy*

$$\langle d, x\rangle = \left\langle A^*(vv^T), x\right\rangle = \left\langle vv^T, Ax\right\rangle = \mathbf{tr}(vv^T Ax) = v^T(Ax)v \leq 0,$$

and for all $y \in \mathcal{F}$ we have:

$$\langle d, y\rangle = \left\langle vv^T, Ay\right\rangle = v^T(Ay)v \geq 0,$$

i.e., $d \in \mathcal{F}^*$, and $\langle d, x\rangle \leq 0$. If (1) has a solution it easily follows that $d \neq 0$ and d can be used in **(b)** *of the perceptron algorithm for a conic system.*

4. Re-scaled Conic Perceptron Algorithm In this section we present and analyze a version of the perceptron algorithm whose complexity depends only logarithmically on $1/\tau_{\mathcal{F}}$. To accomplish this we will systematically re-scale the system (1) using a linear transformation related to a suitably constructed random vector that approximates the center $\bar{z}$ of $\mathcal{F}$. The linear transformation we use was first proposed in [5] for the case of linear inequality systems (i.e., $K = \mathbb{R}^m_+$). Herein we extend these ideas to the conic setting. Table 1 contains a description of our algorithm, which is a structural extension of the algorithm in [5].

Remark 4.1 *In what follows $\mathcal{F}$ denotes the original feasibility cone defined by the initial operator A and the K. By re-scaling on each iteration, the algorithm will define a new linear operator A and the feasibility cone associated with it will be denoted by $\mathcal{F}_A = \{x \in X : Ax \in K\}$.*

Re-scaled Perceptron Algorithm for a Conic System
Step 1 Initialization. Set $B = I$ and $\sigma = 1/(32n)$. Set $J = 0$.
Step 2 Perceptron Algorithm for a Conic System. **(a)** Update iteration counter: $J \leftarrow J + 1$. **(b)** Let x be the origin in X. Repeat at most $\lfloor (1/\sigma^2) \rfloor$ times: **(c)** If $Ax \in \textbf{int}\, K$, Stop. Otherwise, call interior separation oracle for $\mathcal{F}_A$ at x, returning $d \in \mathcal{F}^*_A$, $d \neq 0$, such that $\langle d, x\rangle \le 0$, and set $x \leftarrow x + d/\|d\|$.
Step 3 Stopping Criteria. If $Ax \in \textbf{int}\, K$ then output Bx and **Stop**.
Step 4 Perceptron Improvement Phase. **(a)** Let x be a random unit vector in X. Repeat at most $\lfloor (1/\sigma^2)\ln(n) \rfloor$ times: **(b)** Call deep-separation oracle for $\mathcal{F}_A$ at x with $t = \sigma$. If $\langle d, x\rangle \ge -\sigma\|d\|\|x\|$ for all $d \in \textbf{ext}\mathcal{F}^*_A$ (condition I), End Step 4. Otherwise, oracle returns $d \in \mathcal{F}^*_A$, $d \neq 0$, such that $\langle d, x\rangle \le -\sigma\|d\|\|x\|$ (condition II); set $\bar{d} = d/\|d\|$ and $x \leftarrow x - \langle \bar{d}, x\rangle \bar{d}$. If $x = 0$ restart at **(a)**. **(c)** Call deep-separation oracle for $\mathcal{F}_A$ at x with $t = \sigma$. If oracle returns condition (II), restart at **(a)**.
Step 5 Stopping Criteria. If $Ax \in \textbf{int}\, K$ then output Bx and **Stop**.
Step 6 Re-scaling. $A \leftarrow A\left(I + \frac{xx^T}{\langle x, x\rangle}\right)$, $B \leftarrow B\left(I + \frac{xx^T}{\langle x, x\rangle}\right)$, and Goto **Step 2**.

Table 1: One iteration of the re-scaled perceptron algorithm is one pass of **Steps 2-6**.

The re-scaled perceptron algorithm is initialized in Step 1, after which it passes through Steps 2-6 unless it stops with a solution (in Steps 2, 3, or 5). At Step 6 the matrices A, B are updated and the algorithm loops back to Step 2. The variable J counts the number of times that the algorithm visits Step 2. We consider an "iteration" of the re-scaled perceptron algorithm to be one pass of Steps 2-6 (or, when the algorithm stops with a solution, simply Step 2 through to the stopping step). Thus J counts the number of iterations. We point out for emphasis that J is a random variable.

Note that the perceptron improvement phase (Step 4) requires a deep-separation oracle for $\mathcal{F}_A$ instead of the interior separation oracle for $\mathcal{F}_A$ as required by the perceptron algorithm. For the remainder of this section we presuppose that a deep-separation for $\mathcal{F}_A$ is indeed available. In Section 6 we show how to construct a deep-separation oracle for a variety of useful cones.

We now present our analysis of the re-scaled perceptron algorithm. The following lemma presents intermediate bounds on the width of the feasibility cone $\mathcal{F}_A$, from below, of consecutive iterations of the algorithm. Note in the conclusion of the lemma that the quantity $\|\hat{z}\|$ appears in the denominator; hence the result is intermediate. We will show later, in Lemma 5.2 of Section 5, upper bounds on $\|\hat{z}\|$ and hence more definitive bounds on the width of $\mathcal{F}_A$ over consecutive iterations.

Lemma 4.1 *Let $\bar{z}$ denote the center of the feasibility cone $\mathcal{F}_A$, normalized so that $\|\bar{z}\| = 1$. Let $\hat{A}$ denote the linear operator obtained by re-scaling A in Step 6. Then*

$$\tau_{\mathcal{F}_{\hat{A}}} \geq \frac{(1-\sigma)}{\sqrt{1+3\sigma^2\|\hat{z}\|}}\tau_{\mathcal{F}_A}$$

where $\hat{z} = \bar{z} + \frac{1}{2}\left(\tau_{\mathcal{F}_A} - \left\langle \frac{x}{\|x\|}, \bar{z}\right\rangle\right)\frac{x}{\|x\|}$, and x is the output of the perceptron improvement phase.

Proof. For convenience, let $\tau := \tau_{\mathcal{F}_A}$.At the end of the perceptron improvement phase, we have a vector x satisfying

$$\frac{\langle d, x\rangle}{\|d\|\|x\|} \geq -\sigma \text{ for all } d \in \mathbf{ext}\mathcal{F}_A^*.$$

Let $\bar{x} = x/\|x\|$. Then $\langle d, \bar{x}\rangle \geq -\sigma\|d\|$ for all $d \in \mathbf{ext}\mathcal{F}_A^*$. From Lemma 2.2, it holds that

$$\frac{\langle d, \bar{z}\rangle}{\|d\|\|\bar{z}\|} = \frac{\langle d, \bar{z}\rangle}{\|d\|} \geq \tau \text{ for all } d \in \mathcal{F}_A^*,$$

i.e. $\langle d, \bar{z}\rangle \geq \tau\|d\|$ for all $d \in \mathcal{F}_A^*$. Notice that $A^*\lambda \in \mathcal{F}_A^*$ for all $\lambda \in K^*$, whereby

$$\langle \lambda, A\bar{z}\rangle = \langle A^*\lambda, \bar{z}\rangle \geq \tau_{\mathcal{F}_A}\|A^*\lambda\| \text{ for all } \lambda \in K^*.$$

Note that $\hat{z} = \bar{z} + \frac{1}{2}(\tau - \langle \bar{x}, \bar{z}\rangle)\bar{x}$, and let $\hat{\tau} := \frac{(1-\sigma)}{\sqrt{1+3\sigma^2}}\tau$. We want to show that

$$\langle v, \hat{z}\rangle \geq \hat{\tau}\|v\| \text{ for all } v \in \mathbf{ext}\mathcal{F}_{\hat{A}}^*. \tag{3}$$

If (3) is true, then by convexity of the function $f(v) = \hat{\tau}\|v\| - \langle v, \hat{z}\rangle$ it will also be true that $\langle v, \hat{z}\rangle \geq \hat{\tau}\|v\|$ for any $v \in \mathcal{F}_{\hat{A}}^*$. Then from Lemma 2.2 it would follow that $B(\hat{z}, \hat{\tau}) \subset \mathcal{F}_{\hat{A}}$, whereby $\tau_{\mathcal{F}_{\hat{A}}} \geq \frac{\hat{\tau}}{\|\hat{z}\|}$ as desired.

Let v be an extreme ray of $\mathcal{F}_{\hat{A}}^*$. Using Lemma 2.1, there exists a sequence $\{\lambda^i\}_{i\geq 1}$, $\lambda^i \in K^*$, $\hat{A}^*\lambda^i \to v$ as $i \to \infty$. By Lemma 2.3 we also have that $A^*\lambda^i \to u \in \mathbf{ext}\mathcal{F}_A^*$. Since (3) is trivially true for $v = 0$, we can assume that $v \neq 0$ and hence $A^*\lambda^i \neq 0$ for i large enough. Next note that

$$\|\hat{A}^*\lambda^i\|^2 = \|A^*\lambda^i\|^2 + 2\left\langle A^*\lambda^i, \bar{x}\right\rangle^2 + \langle \bar{x}, \bar{x}\rangle\left\langle A^*\lambda^i, \bar{x}\right\rangle^2 = \|A^*\lambda^i\|^2\left(1 + 3\left(\frac{\langle A^*\lambda^i, \bar{x}\rangle}{\|A^*\lambda^i\|}\right)^2\right)$$

and

$$\begin{array}{rcl} \left\langle \hat{A}^*\lambda^i, \hat{z}\right\rangle & = & \left\langle A^*\lambda^i, \hat{z}\right\rangle + \langle \bar{x}, \hat{z}\rangle\left\langle A^*\lambda^i, \bar{x}\right\rangle \\ & = & \left\langle A^*\lambda^i, \bar{z}\right\rangle + (\tau - \langle \bar{x}, \bar{z}\rangle)\left\langle A^*\lambda^i, \bar{x}\right\rangle + \langle \bar{x}, \bar{z}\rangle\left\langle A^*\lambda^i, \bar{x}\right\rangle \\ & \geq & \tau\|A^*\lambda^i\| + \tau\left\langle A^*\lambda^i, \bar{x}\right\rangle \\ & = & \tau\left(1 + \dfrac{\langle A^*\lambda^i, \bar{x}\rangle}{\|A^*\lambda^i\|}\right)\|A^*\lambda^i\|. \end{array} \tag{4}$$

Therefore $\dfrac{\left\langle \hat{A}^*\lambda^i, \hat{z}\right\rangle}{\|\hat{A}^*\lambda^i\|} \geq \tau\dfrac{1+t_i}{\sqrt{1+3t_i^2}}$ where $t_i = \frac{\langle A^*\lambda^i, \bar{x}\rangle}{\|A^*\lambda^i\|}$. Note that $t_i \leq 1$ and $\langle u, \bar{x}\rangle \geq -\sigma\|u\|$ since $u \in \mathbf{ext}\mathcal{F}_A^*$, and so $\dfrac{\langle u, \bar{x}\rangle}{\|u\|} \geq -\sigma$. By continuity, for any $\varepsilon > 0$ it holds that $t_i \geq -\sigma - \varepsilon$ for i sufficiently large. Thus, $t_i \in [-\sigma - \varepsilon, 1]$ for i large enough.

For $t \in [0, 1]$, we have $\frac{1+t}{\sqrt{1+3t^2}} \geq \frac{1+t}{\sqrt{1+2t+t^2}} = 1$, and for $t \in [-\sigma - \varepsilon, 0]$, the function $g(t) = \frac{1+t}{\sqrt{1+3t^2}} \geq \frac{1-\sigma-\varepsilon}{\sqrt{1+3(\sigma+\varepsilon)^2}}$ since

$$\frac{dg(t)}{dt} = \frac{1-3t}{(1+3t^2)^{3/2}} \geq 0$$

for $t \in [-\sigma - \varepsilon, 0]$, that is, $g(t)$ is increasing on $[-\sigma - \varepsilon, 0]$. Therefore, for i large enough we have

$$\frac{\left\langle \hat{A}\lambda^i, \hat{z} \right\rangle}{\|\hat{A}^*\lambda^i\|} \geq \tau \frac{(1-\sigma-\varepsilon)}{\sqrt{1+3(\sigma+\varepsilon)^2}}.$$

Passing to the limit as $\lambda^i \to v$ we obtain

$$\frac{\langle v, \hat{z} \rangle}{\|v\|} \geq \tau \frac{(1-\sigma-\varepsilon)}{\sqrt{1+3(\sigma+\varepsilon)^2}}$$

whereby

$$\frac{\langle v, \hat{z} \rangle}{\|v\|} \geq \tau \frac{(1-\sigma)}{\sqrt{1+3\sigma^2}} = \hat{\tau}.$$

□

5. Probabilistic Analysis. As mentioned before, the probabilistic analysis of our conic framework is similar to the analysis with linear inequalities in [5]. Although a few changes are required, all the main ideas are still valid. For the sake of completeness, we go over some results of [5]. Our exposition intentionally separates the probabilistic analysis from the remaining sections.

The first lemma of this section was established in [4] for the case of linear inequalities, and here is generalized to the conic framework. Roughly speaking, it shows that the perceptron improvement phase generates near-feasible solutions if started at a good initial point, which happens with at least a fixed probability $p = 1/8$.

Lemma 5.1 *Let z be a feasible solution of (1) of unit norm. With probability at least $\frac{1}{8}$, the perceptron improvement phase (Step 4) visits Step 4(a) only once, and returns a vector x satisfying:*

(i) $\langle d, x \rangle \geq -\sigma\|x\|$ *for every* $d \in \mathbf{ext}\mathcal{F}_A^*$, $\|d\| = 1$, *and*

(ii) $\langle z, x/\|x\| \rangle \geq \frac{1}{\sqrt{n}}$.

Proof. Let x^0 be the random unit vector in $\mathbb{R}^n$ that is the starting value of the perceptron improvement phase. For any given unit vector v, $\mathsf{P}(\left\langle v, x^0 \right\rangle \geq 1/\sqrt{n}) \geq 1/8$; a proof is given in the appendix of [5]. In particular, for the given feasible solution z of unit norm, $\mathsf{P}(\left\langle z, x^0 \right\rangle \geq 1/\sqrt{n}) \geq 1/8$. Notice that in the perceptron improvement phase we have

$$\left\langle x - \left\langle \bar{d}, x \right\rangle \bar{d}, z \right\rangle = \langle x, z \rangle - \left\langle \bar{d}, x \right\rangle \left\langle \bar{d}, z \right\rangle \geq \langle x, z \rangle$$

where $\bar{d} = d/\|d\|$, since $\left\langle \bar{d}, x \right\rangle \leq 0$ and $\left\langle \bar{d}, z \right\rangle \geq 0$ (since $d \in \mathcal{F}_A^*$ and $z \in \mathcal{F}_A$). Thus, $\langle z, x \rangle$ does not decrease at each inner iteration of the perceptron improvement phase (Step 4(b)). Also, in each inner iteration of the perceptron improvement phase (Step 4(b)), the norm of x decreases by at least a constant factor:

$$\begin{aligned} \left\langle x - \left\langle x, \bar{d} \right\rangle \bar{d}, x - \left\langle x, \bar{d} \right\rangle \bar{d} \right\rangle &= \langle x, x \rangle - 2\left\langle \bar{d}, x \right\rangle^2 + \left\langle \bar{d}, x \right\rangle^2 \left\langle \bar{d}, \bar{d} \right\rangle \\ &= \langle x, x \rangle - \left\langle \bar{d}, x \right\rangle^2 = \langle x, x \rangle - \left\langle \bar{d}, x/\|x\| \right\rangle^2 \langle x, x \rangle \\ &\leq \langle x, x \rangle (1 - \sigma^2), \end{aligned}$$

since $\left\langle \bar{d}, x/\|x\| \right\rangle \leq -\sigma < 0$ and $\|\bar{d}\| = 1$.

Thus, after more than $\lfloor (1/\sigma^2)\ln(n) \rfloor$ iterations, we would have $\frac{\langle x, z \rangle}{\|x\|} > 1$, which is a contradiction since z is a unit vector.

Therefore, with probability at least 1/8 we draw a unit random vector x with $\langle z, x \rangle \geq 1/\sqrt{n}$ (so (ii) holds). If this is the case we cannot deeply-separate our point $\lfloor (1/\sigma^2)\ln(n) \rfloor$ times in Step 4(b). So our final point satisfies condition (i).

□

Lemma 5.1 establishes that points obtained after the perceptron improvement phase are near-feasible for the current conic system. The next lemma clarifies the implications of using these near-feasible points to re-scale the conic system.

LEMMA 5.2 *Let A and $\hat{A}$ denote the linear operators of two consecutive iterations of the re-scaled perceptron algorithm. Suppose that $n \geq 2$, $\tau_{\mathcal{F}_A} \leq 1/32n$, and $\sigma \leq 1/32n$. Then*

(i) $\tau_{\mathcal{F}_{\hat{A}}} \geq \left(1 - \frac{1}{32n} - \frac{1}{512n^2}\right)\tau_{\mathcal{F}_A}$;

(ii) With probability at least $\frac{1}{8}$, $\tau_{\mathcal{F}_{\hat{A}}} \geq \left(1 + \frac{1}{3.02n}\right)\tau_{\mathcal{F}_A}$.

PROOF. Let x be the output of the perceptron improvement phase. For simplicity, let $\tau := \tau_{\mathcal{F}_A}$, $\hat{\tau} := \tau_{\mathcal{F}_{\hat{A}}}$, and $\bar{x} = x/\|x\|$. Using Lemma 4.1, we have

$$\hat{\tau} \geq \frac{(1-\sigma)}{\sqrt{1+3\sigma^2}\|\hat{z}\|}\tau \tag{5}$$

where $\hat{z} = \bar{z} + \frac{1}{2}(\tau - \langle \bar{x}, \bar{z}\rangle)\bar{x}$. Next note that

$$\|\hat{z}\|^2 = 1 + (\tau - \langle \bar{x}, \bar{z}\rangle)\langle \bar{x}, \bar{z}\rangle + \frac{1}{4}(\tau - \langle \bar{x}, \bar{z}\rangle)^2 = 1 + \frac{\tau^2}{4} + \langle \bar{z}, \bar{x}\rangle\left(\frac{\tau}{2} - \frac{3}{4}\langle \bar{z}, \bar{x}\rangle\right). \tag{6}$$

Viewing this equation as a quadratic function in $\langle \bar{z}, \bar{x}\rangle$, which is maximized at the value $\langle \bar{z}, \bar{x}\rangle = \tau/3$, we obtain

$$\|\hat{z}\|^2 \leq 1 + \frac{\tau^2}{4} + \frac{\tau^2}{12} = 1 + \frac{\tau^2}{3}.$$

Thus, we have from this inequality and (5) that

$$\hat{\tau} \geq \frac{\tau(1-\sigma)}{\sqrt{1+3\sigma^2}\sqrt{1+\tau^2/3}} \geq \tau(1-\sigma)\left(1 - \frac{3\sigma^2}{2}\right)\left(1 - \frac{\tau^2}{6}\right),$$

where the second inequality uses Proposition 7.1 of the Appendix to bound the square root terms in the denominator. Now invoking the inequalities $\tau, \sigma \leq 1/(32n)$, we have:

$$\begin{aligned}
\hat{\tau} &\geq \tau(1-\sigma)\left(1 - \tfrac{3\sigma^2}{2}\right)\left(1 - \tfrac{\tau^2}{6}\right) \\
&\geq \tau\left(1 - \sigma - \tfrac{3\sigma^2}{2} - \tfrac{\tau^2}{6}\right) \qquad \text{(since } (1-a)(1-b)(1-c) \geq 1-a-b-c \text{ for } a,b,c \geq 0) \\
&\geq \tau\left(1 - \tfrac{1}{32n} - \tfrac{3}{2\times 32^2 n^2} - \tfrac{1}{6\times 32^2 n^2}\right) \\
&= \tau\left(1 - \tfrac{1}{32n} - \tfrac{1}{32^2 n^2}\left(\tfrac{3}{2} + \tfrac{1}{6}\right)\right) \\
&\geq \tau\left(1 - \tfrac{1}{32n} - \tfrac{1}{512n^2}\right).
\end{aligned}$$

Now let us assume that $\langle \bar{z}, \bar{x}\rangle \geq 1/\sqrt{n}$, which happens with probability at least 1/8. In this case, again viewing (6) as a quadratic in $\langle \bar{z}, \bar{x}\rangle$, the quadratic is maximized at $\langle \bar{z}, \bar{x}\rangle = \frac{1}{\sqrt{n}}$, which yields

$$\|\hat{z}\|^2 \leq 1 - \frac{3}{4n} + \frac{\tau}{2\sqrt{n}} + \frac{\tau^2}{4}.$$

Thus, we have from this inequality and (5) that

$$\hat{\tau} \geq \frac{\tau(1-\sigma)}{\sqrt{1+3\sigma^2}\sqrt{1 - \frac{3}{4n} + \frac{\tau}{2\sqrt{n}} + \frac{\tau^2}{4}}} \geq \tau(1-\sigma)\left(1 - \frac{3\sigma^2}{2}\right)\left(1 + \frac{3}{8n} - \frac{\tau}{4\sqrt{n}} - \frac{\tau^2}{8}\right)$$

where the second inequality uses Proposition 7.1 of the Appendix to bound the square root terms in the denominator. Now invoking the inequalities $\tau, \sigma \leq 1/(32n)$, we have:

$$\hat{\tau} \geq \tau\left(1 - \frac{1}{32n}\right)\left(1 - \frac{3}{2048n^2}\right)\left(1 + \frac{3}{8n} - \frac{1}{128n^{1.5}} - \frac{1}{8192n^2}\right) \geq \tau\left(1 + \frac{1}{3.02n}\right),$$

where the last inequality follows from Proposition 7.2 of the Appendix.

□

The following theorem bounds the number of iterations and the number of oracle calls made by the re-scaled perceptron algorithm. Recall that an iteration is one pass of Steps 2-6, and the variable J in the algorithm counts the number of iterations.

THEOREM 5.1 *Suppose that $n \geq 2$ and (1) has a solution, and that $\delta \in (0,1)$ is given. Then, with probability at least $1-\delta$, the re-scaled perceptron algorithm will compute a solution of (1) in no more than*

$$T = \max\left\{4096\ln\left(\frac{1}{\delta}\right), 139n\ln\left(\frac{1}{32n\tau_{\mathcal{F}}}\right)\right\} = O\left(n\ln\left(\frac{1}{\tau_{\mathcal{F}}}\right)+\ln\left(\frac{1}{\delta}\right)\right)$$

iterations. Moreover, with probability at least $1-\delta$, the algorithm makes at most $O(T\ n^2\ \ln(n))$ calls of the deep-separation oracle for $\mathcal{F}_A$ and at most $O(T\ n^2)$ calls of the interior separation oracle for $\mathcal{F}_A$.

PROOF. Our proof is slightly different than that of Theorem 3.4 in [5]. Let A denote the event $\{J > T\}$, where J is the total number of iterations of the re-scaled perceptron algorithm (i.e., visits to Step 2). Then to prove the theorem we must show that $\mathsf{P}(A) \leq \delta$. We proceed as follows. Let U denote the total number of times that the re-scaled perceptron algorithm calls Step 4(a), hence $U \geq J-1$, and let i index these calls. After each visit to Step 4(a) exactly one of three cases can occur: (i) the algorithm ends Step 4 with the resulting update in Step 6 satisfying conclusion (ii) of Lemma 5.1, (ii) the algorithm ends Step 4 with the resulting update in Step 6 not satisfying the conclusion (ii) of Lemma 5.1, or (iii) the algorithm does not end Step 4 and therefore restarts Step 4(a). For $i = 1,\dots,U$, let V_i be the binary random variable whose value is 1 if the i^{th} call of Step 4(a) ends in case (i), and is 0 otherwise. Lemma 5.2 implies that $\mathsf{P}(V_i = 1) \geq 1/8$. If $U < T$, define V_i for $i = U+1,\dots,T$ to be a Bernoulli random variable that takes values 1 and 0 with probabilities 1/8 and 7/8, respectively. Now define $V = \sum_{i=1}^{T} V_i$, and it follows that $E[V] \geq T/8$. Let B denote the event $\{V < \frac{15}{16}E[V]\}$. Applying a Chernoff bound (see Theorem 4.2 of [13]) with $\varepsilon = 1/16$, we have:

$$\mathsf{P}(B) = \mathsf{P}(V < (1-\varepsilon)E[V]) < e^{-\varepsilon^2 E[V]/2} = e^{-E[V]/512} \leq e^{-T/4096} \leq \delta\ ,$$

since $E[V] \geq T/8$ and $T \geq 4096\ln(1/\delta)$.

Now note that

$$\mathsf{P}(A) = \mathsf{P}(A\cap B) + \mathsf{P}(A\cap B^c) \leq \mathsf{P}(B) + \mathsf{P}(A\cap B^c) < \delta + \mathsf{P}(A\cap B^c)\ .$$

Therefore the theorem will be proved if we can show that $\mathsf{P}(A\cap B^c) = 0$, which we now do by contradiction.

Indeed, suppose that events $\{U > T\}$ and B^c are realized (note that $A \subset \{U > T\}$), then we have $U > T$ and $V \geq \frac{15}{16}E[V]$. For $i = 0,\dots,T$, let τ_i denote the width of the feasibility cone after i calls to Step 4(a) (hence $\tau_0 = \tau_{\mathcal{F}}$). For all $i = 0,\dots,U-1$ it holds that $1/(32n) > \tau_i$ (otherwise Step 2(b) finds a feasible solution and the algorithm stops), whereby $1/(32n) > \tau_T$ since $T < U$. It follows from Lemma 5.2 that $V_i = 1$ implies $\tau_{i+1} \geq \tau_i(1+1/(3.02n))$. If $V_i = 0$, then either case (ii) or case (iii) above occur, the former yielding $\tau_{i+1} \geq \tau_i\left(1-\frac{1}{32n}-\frac{1}{512n^2}\right)$ from Lemma 5.2, and the latter yielding $\tau_{i+1} = \tau_i$ (i.e., no update is performed). Therefore

$$\begin{array}{rcl}\tau_T & \geq & \tau_0\left(1+\frac{1}{3.02n}\right)^{V}\left(1-\frac{1}{32n}-\frac{1}{512n^2}\right)^{T-V}\\ & \geq & \tau_{\mathcal{F}}\left(1+\frac{1}{3.02n}\right)^{15E[V]/16}\left(1-\frac{1}{32n}-\frac{1}{512n^2}\right)^{T-15E[V]/16}\\ & \geq & \tau_{\mathcal{F}}\left(1+\frac{1}{3.02n}\right)^{\frac{15T}{128}}\left(1-\frac{1}{32n}-\frac{1}{512n^2}\right)^{T-\frac{15T}{128}}\\ & = & \tau_{\mathcal{F}}\left[\left(1+\frac{1}{3.02n}\right)^{\frac{15}{128}}\left(1-\frac{1}{32n}-\frac{1}{512n^2}\right)^{\frac{113}{128}}\right]^{T}\\ & \geq & \tau_{\mathcal{F}}e^{T/139n}\\ & \geq & 1/(32n)\ ,\end{array}$$

where the second-to-last inequality follows from Proposition 7.4 of the Appendix, and the last inequality follows since $T \geq 139n\ln(1/(32n\tau_{\mathcal{F}}))$. This contradicts the fact that $1/(32n) > \tau_T$ which was shown above; hence $\mathsf{P}(A\cap B^c) \leq \mathsf{P}(\{U > T\}\cap B^c) = 0$ and the main bound is proven. It follows that $J \leq U \leq T$ with probability at least $1-\delta$. Therefore, with probability at least $1-\delta$, the number of calls to the separation oracle for $\mathcal{F}_A$ (Step 2) is at most $\lfloor 1024n^2T\rfloor$ and the number of calls to the deep-separation oracle for $\mathcal{F}_A$ (Step 4(a)) is at most $\lfloor 1024n^2\ln(n)\ T\rfloor$. □

REMARK 5.1 *It is instructive to compare the complexity bound in Theorem 5.1 with that of the ellipsoid method (see [10]). Let W_s and W_d denote the number of operations needed for an oracle call to an interior separation oracle and a deep-separation oracle, respectively, for the feasibility cone $\mathcal{F}$ (or $\mathcal{F}_A$). The complexity of the ellipsoid method for computing a solution of (1) is $O(n^2\ln(1/\tau_{\mathcal{F}}))$ iterations, with each iteration requiring (i) one call to an interior separation oracle for $\mathcal{F}$, and (ii) $O(n^2)$ additional operations,* *yielding a total operation count of $O((n^4 + n^2 W_s)\ln(1/\tau_{\mathcal{F}}))$. The corresponding complexity bound for the re-scaled perceptron algorithm is $O(n\ln(1/\tau_{\mathcal{F}}) + \ln(1/\delta))$ iterations, where each iteration requires (i) $O(n^2)$ calls to an interior separation oracle, (ii) $O(n^2 \ln n)$ calls to a deep-separation oracle, and $O(n^2)$ additional operations, yielding a total operation count of $O((n^2W_s+n^2\ln nW_d+n^2)(n\ln(1/\tau_{\mathcal{F}})+\ln(1/\delta)))$. If we make the reasonable presumption that either δ is a fixed scalar or $\tau_{\mathcal{F}} << \delta$, and that $W_d \geq W_s$, we see that the ellipsoid method has superior complexity by a factor of at least $n \ln n$, with this advantage growing to the extent that $W_d >> W_s$ (as is the case when K is either composed of second-order or positive semi-definite cones, see Section 6). However, the re-scaled perceptron algorithm is still attractive for at least two reasons. First, it has the possibility of acceleration beyond its worst-case bound. And second, we believe that the method is of independent interest for its ability to re-dilate the space in a way that improves the width of the feasibility cone. It may be possible to exploit the mechanisms underlying this phenomenon in other algorithms yet to be developed.*

6. Deep-separation Oracles for $\mathcal{F}_A$ and their Extensions, for Some Inclusion Cones K

The re-scaled perceptron algorithm presupposes the availability of a deep-separation oracle for the feasibility cone $\mathcal{F}_A$. Herein we discuss how to construct such a deep separation oracle for $\mathcal{F}_A$ for certain inclusion cones and their cross-products. Before doing so, we first extend the concept of a deep-separation oracle in two ways.

Definition 6.1 *For a fixed positive scalar t and a fractional value $\alpha \in (0,1]$, an* α-deep-separation oracle *for a cone $C \subset \mathbb{R}^n$ is a subroutine that given a non-zero point $x \in \mathbb{R}^n$, either*

(I) correctly identifies that $\dfrac{\langle d, x\rangle}{\|d\|\|x\|} \geq -t$ *for all* $d \in \mathbf{ext}C^*$

or

(II) returns a vector $d \in C^*$, $d \neq 0$, satisfying $\dfrac{\langle d, x\rangle}{\|d\|\|x\|} \leq -\alpha t.$

Definition 6.1 only differs from Definition 2.3 in the inequality in condition (II), where now αt is used instead of t. In order to properly use this relaxed oracle, it is only required to modify the iteration bound used in Step 4(a) of the re-scaled perceptron algorithm as follows:

Step 4(a) Let x be a random unit vector in X. Repeat at most $\lfloor (1/(\alpha^2\sigma^2))\ln(n) \rfloor$ times:

For example, by setting $\alpha = 1/2$ in the constructions later in this section will therefore increase the iteration bound of Step 4 by a constant factor; all other analysis remains valid with no modifications. We also extend the concept of a deep-separation oracle probabilistically as follows.

Definition 6.2 *For a fixed positive scalar t, a fractional value $\alpha \in (0,1]$, and a probability of failure γ, an* (α,γ)-deep-separation oracle *for a cone $C \subset \mathbb{R}^n$ is a subroutine that given a non-zero point $x \in \mathbb{R}^n$, either*

(I) identifies that " $\dfrac{\langle d, x\rangle}{\|d\|\|x\|} \geq -t$ *for all* $d \in \mathbf{ext}C^*$*" holds with probability at least $1-\gamma$*

or

(II) returns a vector $d \in C^*$, $d \neq 0$, satisfying $\dfrac{\langle d, x\rangle}{\|d\|\|x\|} \leq -\alpha t.$

Definition 6.2 only differs from Definition 6.1 in the correctness of assertion (I), where now this assertion is incorrect with probability at most γ.

We now discuss how to modify the re-scaled perceptron algorithm to utilize an (α,γ)-deep-separation oracle with associated complexity bounds similar to those of Theorem 5.1. For a given overall probability of failure δ and value of $\sigma = 1/(32n)$, let $i = 1,\ldots,$ index the visits of the re-scaled perceptron algorithm to Step 4(a). For each i, pre-assign a probability of failure $\gamma = p_i$ for the (α,γ)-deep-separation oracle called in Steps 4(b) and/or 4(c) immediately following the i^{th} visit to Step 4(a), as follows:

$$p_i = \frac{3\delta\alpha^2\sigma^2}{10 i^2 \ln(n)} \ .$$

Noting that each visit to Step 4(a) results in at most $\lfloor (1/(\alpha^2\sigma^2))\ln(n)\rfloor$ calls of the (α,γ)-deep-separation oracle, the probability that one or more of these oracle calls associated with the i^{th} visit to Step 4(a) will return an incorrect assertion is at most:

$$(1/(\alpha^2\sigma^2))\ln(n)p_i \le \frac{3\delta}{10i^2} \ .$$

Therefore, over all iterations of the re-scaled perceptron algorithm, the probability that one or more (α,γ)-deep-separation oracle calls will return an incorrect assertion is at most:

$$\sum_{i=1}^{U}\frac{3\delta}{10i^2} \le \sum_{i=1}^{\infty}\frac{3\delta}{10i^2} = \frac{3\delta\pi^2}{10\times 6} < \frac{\delta}{2} \ ,$$

where the random variable U is the total number of times that the re-scaled perceptron algorithm calls Step 4(a), and the infinite series equality for $\pi^2/6$ is well-known (see page 8 of [9]). Therefore, the probability of failure of the re-scaled perceptron algorithm due to incorrect assertions of the (α,γ)-deep-separation oracle is at most $\delta/2$. The other source of failure of the re-scaled perceptron algorithm is due to the possibility that sufficiently many initial random unit vectors x chosen in Step 4(a) will not satisfy the cosine inequality $\langle z,x\rangle \ge 1/\sqrt{n}$. But as the analysis in Section 5 showed (specifically Theorem 5.1), the probability of failure of the re-scaled perceptron algorithm due to failure of enough iterations to satisfy the cosine inequality is at most $\delta/2$ if the algorithm is run for

$$\hat{T} = \left\lceil \max\left\{4096\ln\left(\frac{1}{\delta/2}\right), 139n\ln\left(\frac{1}{32n\tau_{\mathcal{F}}}\right)\right\}\right\rceil = O\left(n\ln\left(\frac{1}{\tau_{\mathcal{F}}}\right)+\ln\left(\frac{1}{\delta}\right)\right)$$

iterations, which is of the same order as the bound in Theorem 5.1. Therefore we can achieve the same order complexity bound using an (α,γ)-deep-separation oracle as with the other deep-separation oracles of Definitions 2.3 or 6.1.

The computational cost of using low values of failure probabilities $\gamma = p_i$ in the (α,γ)-deep-separation oracle bears further scrutiny. Let us presume (as will be the case in our application of the (α,γ)-deep-separation oracle in Section 6.4) that the complexity of running the (α,γ)-deep-separation oracle on γ is $O(\ln(1/\gamma))$ where the constants may depend on other problem scalars but not on γ, i.e., the complexity grows at most linearly in $\ln(1/\gamma)$. The largest computational cost of any of the calls of the (α,γ)-deep-separation oracle will be bounded by $O(\max_i \ln(1/p_i)) = O(\ln(1/p_U))$ where again the random variable U is the total number of times that the re-scaled perceptron algorithm calls Step 4(a). Re-tracing the steps of the proof of Theorem 5.1 but using $\hat{T}$ instead of T, one can show that the re-scaled perceptron algorithm will compute a solution of (1) in no more than $\hat{T}$ visits to Step 4(a) (i.e., $U \le \hat{T}$) with probability at least $1-\delta$. Therefore, with probability $1-\delta$ the algorithm is successful within $\hat{T}$ iterations and furthermore $U \le \hat{T}$, and hence the computational cost of any call to the (α,γ)-deep-separation oracle will be at most

$$\begin{aligned} O(\ln(1/p_{\hat{T}})) &\le O\left(\ln\left(\tfrac{10\hat{T}^2\ln(n)}{3\delta\sigma^2\alpha^2}\right)\right) \\ &\le O\left(\ln\left(\tfrac{n\hat{T}}{\delta\alpha}\right)\right) \\ &\le O\left(\ln\left(\tfrac{n\ln(1/\tau_{\mathcal{F}})}{\delta\alpha}\right)\right) \ . \end{aligned}$$

For the rest of this section, the term "deep-separation oracle" will refer to either Definition 2.3, 6.1, or 6.2, where the particular definition will be clear from context.

We consider instances of (1) that are themselves intersections of conic inclusions of families of the three canonical inclusion cones of modern convex optimization: the nonnegative orthants $\mathbb{R}^l_+$, the second-order cones Q^k, and the semidefinite cones $S^{k\times k}_+$. We consider a specific instance to contain some subset of the following inclusions (but trivially must contain the fifth inclusion):

$$\left\{\begin{array}{rcll} A_Lx &\in& \textbf{int}\ \mathbb{R}^l_+ & \\ A_{Q_i}x &\in& \textbf{int}\ Q^{n_i} & i=1,\dots,q \\ Ix &\in& \textbf{int}\ S^{p\times p}_+ & \\ A_{S_i}x &\in& \textbf{int}\ S^{k_i\times k_i}_+ & i=1,\dots,s \\ x &\in& X\ . & \end{array}\right. \tag{7}$$

Note that we distinguish the third conic inclusion in (7) from the fourth more general semidefinite inclusion since the linear operator in the former is the identity operator. We will show below that this third inclusion has particularly nice structure for a deep-separation oracle and for the re-scaled perceptron algorithm itself. Note that (7) is an instance of (1) for $K = \mathbb{R}^l_+ \times Q^{n_1} \times \cdots \times Q^{n_q} \times S^{p\times p}_+ \times S^{k_1\times k_1}_+ \times \cdots \times S^{k_s\times k_s}_+$. The starting point of our analysis is a simple observation about intersections of feasibility cones. Suppose we have available deep-separation oracles for each of the feasibility cones $\mathcal{F}_1$ and $\mathcal{F}_2$ of instances:

$$\left\{\begin{array}{rcl} A_1x & \in & \textbf{int}\ K_1 \\ x & \in & X \end{array}\right. \quad \text{and} \quad \left\{\begin{array}{rcl} A_2x & \in & \textbf{int}\ K_2 \\ x & \in & X \end{array}\right. \tag{8}$$

and consider the problem of finding a point x that satisfies both conic inclusions:

$$\left\{\begin{array}{rcl} A_1x & \in & \textbf{int}\ K_1 \\ A_2x & \in & \textbf{int}\ K_2 \\ x & \in & X\ . \end{array}\right. \tag{9}$$

Let $\mathcal{F}_A = \{x : A_1x \in K_1, A_2x \in K_2\} = \{x : Ax \in K\}$ where $K = K_1 \times K_2$ and A is defined analogously. Then $\mathcal{F}_A = \mathcal{F}_1 \cap \mathcal{F}_2$ where $\mathcal{F}_i = \{x : A_ix \in K_i\}$ for $i = 1, 2$. It follows from the calculus of convex cones that $\mathcal{F}^*_A = \mathcal{F}^*_1 + \mathcal{F}^*_2$, and therefore

$$\textbf{ext}\mathcal{F}^*_A \subset (\textbf{ext}\mathcal{F}^*_1 \cup \textbf{ext}\mathcal{F}^*_2)\ . \tag{10}$$

This observation leads to an easy construction of a deep-separation oracle for $\mathcal{F}_A = \mathcal{F}_1 \cap \mathcal{F}_2$ if one has available (α, γ_i)-deep-separation oracles for $\mathcal{F}_i$ for $i = 1, 2$:

$(\alpha, \gamma_1 + \gamma_2)$**-Deep-separation Oracle for** $\mathcal{F}_1 \cap \mathcal{F}_2$
Given: scalar $t > 0$ and $x \neq 0$, call the (α, γ_i)-deep-separation oracles for $\mathcal{F}_i$, $i = 1, 2$, at x.
If both oracles assert Condition I, then assert Condition I.
Otherwise at least one oracle reports Condition II and provides $d \in \mathcal{F}^*_i \subset \mathcal{F}^*_A$, $d \neq 0$, such that $\langle d, x\rangle \leq -\alpha t\|d\|\|x\|$; return d and Stop.

Remark 6.1 *If (α, γ_i)-deep-separation oracles for $\mathcal{F}_i$ are available and their complexity is $O(T_i)$ operations for $i = 1, 2$, then the oracle for $\mathcal{F}_1 \cap \mathcal{F}_2$ given above is an $(\alpha, \gamma_1 + \gamma_2)$-deep-separation oracle, and its complexity is $O(T_1 + T_2)$ operations.*

Utilizing Remark 6.1, in order to construct a deep-separation oracle for the feasibility cone of (7) it will suffice to construct deep-separation oracles for each of the conic inclusions therein, which is what we now examine.

6.1 Deep-separation Oracle for $\mathcal{F}_A$ when $K = \mathbb{R}^m_+$ We consider $\mathcal{F}_A = \{x : Ax \in \mathbb{R}^m_+\}$. Example 2.2 has already described a deep-separation oracle for $\mathcal{F}_A$ when the inclusion cone is $\mathbb{R}^m_+$. It is easy to see that this oracle can be implemented in $O(mn)$ operations.

6.2 (1/2)-Deep-separation Oracle for $\mathcal{F}_A$ when $K = Q^k$ For convenience we amend our notation so that $\mathcal{F}_A = \{x : \|Mx\| \leq g^Tx\}$ for a given real $(k-1)\times n$ matrix M and a real n-vector g, so that $\mathcal{F}_A = \{x : Ax \in Q^k\}$ where the linear operator A is specified by $Ax := \begin{bmatrix} Mx \\ g^Tx \end{bmatrix}$. We will construct an efficient $\alpha = (1/2)$-deep-separation oracle (Definition 6.1) by considering the following optimization problem:

$$\begin{array}{rcll} t^* & := & \min_d & d^Tx \\ & & \text{s.t.} & \|d\| = 1 \\ & & & d \in \mathcal{F}^*_A\ . \end{array} \tag{11}$$

If $x \in \mathcal{F}_A$, then $t^* \geq 0$ and clearly condition I of Definition 6.1 is satisfied. If $x \notin \mathcal{F}_A$, then $t^* < 0$ and we can replace the equality constraint in (11) with an inequality constraint. We obtain the following primal/dual pair of convex problems with common optimal objective function value t^*:

$$
\begin{array}{rllcrl}
t^* := \min_d & x^T d & & = & \max_y & -\|y-x\| \\
\text{s.t.} & \|d\| \leq 1 & & & \text{s.t.} & y \in \mathcal{F}_A \\
& d \in \mathcal{F}_A^* & & & &
\end{array} \tag{12}
$$

Now consider the following (1/2)-deep-separation oracle for $\mathcal{F}_A$ when $K = Q^k$.

(1/2)-Deep-Separation Oracle for $\mathcal{F}_A$ when $K = Q^k$, for $x \neq 0$ and parameter $t > 0$
If $\|Mx\| \leq g^T x$, return Condition I, and Stop.
Solve (12) for feasible primal and dual solutions $\bar{d}, \bar{y}$ with duality gap $\bar{g}$ satisfying $\bar{g}/\|x\| \leq t/2$
If $x^T\bar{d}/\|x\| \geq -t/2$, report Condition (I), and Stop.
If $x^T\bar{d}/\|x\| \leq -t/2$, then return $d = \bar{d}$, report Condition (II), and Stop.

To see the validity of this method, note that if $\|Mx\| \leq g^T x$, then $x \in \mathcal{F}_A$ and clearly Condition (I) of Definition 6.1 is satisfied. Next, suppose that $x^T\bar{d}/\|x\| \geq -t/2$, then $t^* \geq -\|\bar{y} - x\| = x^T\bar{d} - \bar{g} \geq -\|x\|t/2 - \|x\|t/2 = -\|x\|t$. Therefore $\frac{x^T d}{\|x\|\|d\|} \geq -t$ for all $d \in \mathcal{F}_A^*$, and it follows that Condition (I) of Definition 6.1 is satisfied. Finally, if $x^T\bar{d}/\|x\| \leq -t/2$, then $\frac{\bar{d}^T x}{\|\bar{d}\|\|x\|} \leq -t/2$ and $\bar{d} \in \mathcal{F}_A^*$, whereby Condition (II) of Definition 6.1 is satisfied using $\bar{d}$.

The computational complexity of this deep-separation oracle depends on the ability to efficiently solve (12) for feasible primal/dual solutions with duality gap $\bar{g} \leq t\|x\|/2$. For the case when $K = Q^k$, it is shown in [1] that (12) can be solved very efficiently to this desired duality gap, namely in $O(mn^2 + n\ln\ln(1/t) + n\ln\ln(1/\min\{\tau_{\mathcal{F}_A}, \tau_{\mathcal{F}_A^*}\}))$ operations, using a combination of Newton's method and binary search. Using $t = \sigma := 1/(32n)$ this is $O(mn^2 + n\ln\ln(1/\min\{\tau_{\mathcal{F}_A}, \tau_{\mathcal{F}_A^*}\}))$ operations for the relaxation parameter σ needed by the re-scaled perceptron algorithm.

6.3 Deep-separation Oracle for $\mathcal{F} = S_+^{p\times p}$ We consider the instance of (7) that contains the third conic inclusion " $Ix \in S_+^{p\times p}$ " and for convenience we temporarily alter our notation herein so that $X \in S^{p\times p}$ is a point under consideration. A deep-separation oracle for $S_+^{p\times p}$ at $X \neq 0$ for the scalar $t > 0$ is constructed by simply checking the condition "$X + t\|X\|I \succeq 0$." If $X + t\|X\|I \succeq 0$, then condition I of the deep-separation oracle is satisfied. This is true because the extreme rays of $\mathcal{F}^* = (S_+^{p\times p})^* = S_+^{p\times p}$ are the collection of rank-1 matrices vv^T, and

$$\frac{\langle vv^T, X\rangle}{\|X\|\|vv^T\|} = \frac{v^T X v}{\|X\|\|vv^T\|} \geq \frac{-t\|X\|v^T v}{\|X\|\|vv^T\|} = -t$$

for any $v \neq 0$. On the other hand, if $X + t\|X\|I \not\succeq 0$, then compute any v satisfying $v^T X v + t\|X\|v^T v < 0$, and return $D = vv^T$, which will satisfy

$$\frac{\langle D, X\rangle}{\|X\|\|D\|} = \frac{v^T X v}{\|X\|v^T v} \leq -t\ ,$$

thus satisfying condition II. Notice that the work per oracle call is simply to check the eigenvalue condition $X + t\|X\|I \succeq 0$ and possibly to compute an appropriate vector v.

In order to convey the key insights of the analysis, throughout this section we assume that we can compute exactly the minimum eigenvalue and an associated eigenvector. We defer to the appendix the complete complexity analysis when we can only approximate these quantities.

While a deep-separation oracle for $S_+^{p\times p}$ is straightforward to construct and hence can be used for the third conic inclusion " $Ix \in S_+^{p\times p}$ " of (7), the re-scaling step (Step 6) of the re-scaled perceptron algorithm will modify this inclusion to " $Bx \in S_+^{p\times p}$ " for some matrix B in Step 6 of the first iteration as well as at Step 6 of all subsequent iterations, thus destroying the special structure of the inclusion that led to the simple deep-separation oracle described above. However, it turns out that the re-scaled perceptron algorithm can be slightly modified to handle the inclusion " $Ix \in S_+^{p\times p}$ " *without* a deep-separation oracle for this inclusion, i.e., using only a deep-separation oracle for the other inclusions in the problem instance. For ease of exposition we return to our standard notation denoting our variable by x, etc., and we write our instance of (7) as

$$\begin{cases} A_1 x & \in & \textbf{int } K_1 \\ Ix & \in & \textbf{int } K_2 \\ x & \in & X \ , \end{cases} \tag{13}$$

where $K_2 := S_+^{p\times p}$, and note that K_2 is self-dual. Suppose that we have run $k+1$ iterations of the re-scaled perceptron algorithm for this problem instance, and let $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{k+1}$ denote the iterate sequence of normalized ($\|\bar{x}_i\| = 1$) output vectors x at the end of Step 4 of the re-scaled perceptron algorithm, yielding the re-scaling matrices $B_0 = I, B_1, B_2, \ldots, B_k$, where $B_i = (I + \bar{x}_1\bar{x}_1^T)(I + \bar{x}_2\bar{x}_2^T)\cdots(I + \bar{x}_i\bar{x}_i^T)$, $i = 1, \ldots, k$. Here $\bar{x}_{k+1}$ is the output based on the re-scaling matrix B_k and the re-scaled problem instance:

$$\begin{cases} A_1 B_k x & \in & \textbf{int } K_1 \\ B_k x & \in & \textbf{int } K_2 \\ x & \in & X \ . \end{cases} \tag{14}$$

From the perceptron improvement phase (Step 4), we have no guarantee that $B_k\bar{x}_{k+1} \in K_2$. However, if such is the case, we have the following result which will be useful algorithmically:

Lemma 6.1 *Suppose that K_2 is self-dual, and that $B_i\bar{x}_{i+1} \in K_2$ and $\|\bar{x}_{i+1}\| = 1$ for $i = 0, \ldots, k$. Then $B_iB_i^* d \in K_2$ for all $d \in K_2^*$ for $i = 0, \ldots, k+1$.*

Proof. We proceed by induction on i. Since $B_0 = B_0^* = I$ the statement trivially holds for $i = 0$ due to the self-duality of K_2. Next assume the statement is true for a given $i \le k$. Therefore for all $d \in K_2^*$ we have:

$$\begin{aligned} B_{i+1}B_{i+1}^* d &= B_i(I + \bar{x}_{i+1}\bar{x}_{i+1}^T)(I + \bar{x}_{i+1}\bar{x}_{i+1}^T)B_i^* d \\ &= B_iB_i^* d + 3B_i\bar{x}_{i+1}\bar{x}_{i+1}^T B_i^* d \in K_2 \end{aligned}$$

by the induction assumption and the hypothesis that $B_i\bar{x}_{i+1} \in K_2$. □

Lemma 6.1 states that if every point $\bar{x}_{i+1}$ used to re-scale A satisfies $B_i\bar{x}_{i+1} \in K_2$, we have that $B_iB_i^*$ maps the semidefinite cone into itself. As we show below this allows for updates on x that removes the need of a deep-separation oracle but does not harm the complexity analysis.

In order to take advantage of Lemma 6.1, we now show how to modify the perceptron improvement phase (Step 4) of the re-scaled perceptron algorithm to guarantee that $B_i\bar{x}_{i+1} \in K_2$ for all i. The deep-separation oracle for $\{x : B_k x \in K_2\}$ is replaced by the following "update algorithm".

"Update Algorithm"
Given the current iterate x:
Step 1 If $B_k x \notin K_2 (= S_+^{p\times p})$
  Step 1(a) Let $d = vv^T \in K_2$ where v is an eigenvector of a negative eigenvalue of $B_k x$
  Step 1(b) Set $\theta := -\langle d, B_k x\rangle / \|B_k^* d\|^2$ and update the iterate as $x^+ \leftarrow x + \theta B_k^* d$
  Step 1(c) Set $x \leftarrow x^+$ and goto Step 1.
Step 2 Report x.

Lemma 6.2 *Replacing the deep-separation oracle for $\{x : B_k x \in K_2\}$ by the "Update Algorithm" does not affect the computational complexity of the outer loops of the re-scaled perceptron algorithm.*

Proof. Suppose that we have just completed k iterations of the re-scaled perceptron algorithm, and let z^k denote the center of the re-scaled problem instance (14). Suppose that we are now in Step 4 of iteration $k+1$. Let us examine the case where the starting vector x of Step 4(a) satisfies $\langle z^k, x\rangle / \|x\| \ge 1/\sqrt{n} > 0$ (which happens with probability at least 1/8). In this case consider any x generated in Step 4(b). If $B_k x \notin K_2 (= S_+^{p\times p})$, we let $d = vv^T \in K_2$ where v is an eigenvector of a negative eigenvalue of $B_k x$, and replace

$$x^+ \leftarrow x + \theta B_k^* d \tag{15}$$

where $\theta := -\langle d, B_k x\rangle / \|B_k^* d\|^2$. It then follows that

$$\langle x^+, z^k\rangle = \langle x, z^k\rangle + \theta\langle B_k^* d, z^k\rangle \ge \langle x, z^k\rangle$$

since $B_k z^k \in K_2$ and hence $\langle d, B_k z^k \rangle \geq 0$. Furthermore, from the particular choice of θ we have

$$\|x^+\|^2 = \|x\|^2 + 2\theta \langle x, B_k^* d\rangle + \theta^2 \|B_k^* d\|^2 = \|x\|^2 - \frac{\langle d, B_k x\rangle^2}{\|B_k^* d\|^2} \leq \|x\|^2 ,$$

and hence the potential function $\langle z^k, x\rangle / \|x\|$ is non-decreasing if we replace x by x^+. If all previous iterates satisfied $B_i x_{i+1} \in K_2$, $i = 1, \ldots, k-1$, then we have from Lemma 6.1 (using $k-1$ instead of k) that

$$B_k x^+ - B_k x = \theta B_k B_k^* d \in K_2$$

and furthermore from the choice of θ we have

$$\begin{aligned} v^T(B_k x^+)v &= v^T(B_k x + \theta B_k B_k^* d)v = \langle B_k x + \theta B_k B_k^* d, vv^T\rangle \\ &= \langle B_k x + \theta B_k B_k^* d, d\rangle = 0 \ . \end{aligned}$$

Therefore $B_k x^+ \succeq B_k x$ and $B_k x^+$ has at least one fewer negative eigenvalue than $B_k x$. It follows that after repeating the replacement at most p times we ensure that the final replacement value x^+ satisfies $B_k x^+ \in K_2$. Inductively this shows that we can run the perceptron improvement phase generating values of x^+ that satisfy $B_k x^+ \in K_2$ whose potential function value for the perceptron improvement methodology is improved. Therefore there is no need for a deep-separation oracle for the feasibility cone $\mathcal{F}_2 = \{x : B_k x \in K_2\}$ and it suffices only to have a deep-separation oracle for the feasibility cone $\mathcal{F}_1 = \{x : A_1 B_k x \in K_1\}$.

This modified version of the perceptron improvement phase has the same internal iteration bound (repeat at most $\lfloor \ln(n)/\sigma^2 \rfloor$ Step 4(b)) and therefore leaves unchanged the overall complexity bound of Theorem 5.1 for the re-scaled perceptron algorithm. (Note that the complexity bound of the proposed oracle, which incorporates possibly many updates of the form (15), will also depend on p but the oracle does not affect the number of outer iterations T.) □

6.4 $(1/2, \delta)$-Deep-separation Oracle for $\mathcal{F}_A$ when $K = S_+^{p\times p}$ In this subsection we present a methodology for a $(1/2, \delta)$-deep-separation oracle for $\mathcal{F}_A = \{x : Ax \in S_+^{p\times p}\}$ for the conic system:

$$\left\{ \begin{array}{rcl} Ax & \in & \textbf{int}\ S_+^{p\times p} \\ x & \in & X \ . \end{array} \right. \tag{16}$$

Our analysis uses the data-perturbation condition measure model of Renegar [17], which we now briefly review. Considering (16) as a system with fixed cone $K = S_+^{p\times p}$ and fixed spaces X and Y, then $A \in L(X,Y)$ where $L(X,Y)$ is the space of linear operators from X to Y. Let $\|\cdot\|_O$ denote the operator norm on $L(X,Y)$, namely $\|V\|_O := \max_{0\neq x\in X} \|Vx\|/\|x\|$ for $V \in L(X,Y)$, where the norms on X and Y are the inner product norms. Let $\mathcal{M} \subset L(X,Y)$ denote those linear operators $A \in L(X,Y)$ for which (1) has a solution. For $A \in \mathcal{M}$, let $\rho(A)$ denote the "distance to infeasibility" for (1), namely:

$$\rho(A) := \min_{\Delta A} \{\|\Delta A\|_O : A + \Delta A \notin \mathcal{M}\} \ .$$

Then $\rho(A)$ denotes the smallest perturbation of our given operator A which would render the system (1) infeasible. Next let $\mathcal{C}(A)$ denote the *condition measure* of (1), namely $\mathcal{C}(A) = \|A\|_O/\rho(A)$, which is a scale-invariant reciprocal of the distance to infeasibility. We note that $\ln(\mathcal{C}(A))$ is tied to the complexity of interior-point methods and the ellipsoid method for computing a solution of (1), see [18] and [6].

Given the inclusion cone $K = S_+^{p\times p}$, the feasibility cone for (1) is $\mathcal{F}_A = \{x : Ax \in K\}$. Given the relaxation parameter $t > 0$ and a non-zero vector $x \in \mathbb{R}^n$, consider the following conic feasibility system in the variable w:

$$(S_{t,x}) : \left\{ \begin{array}{rcl} t\|x\|\|A^* w\| + \langle w, Ax\rangle & < & 0 \\ \\ w & \in & \textbf{int}\ S_+^{p\times p} \end{array} \right. \tag{17}$$

Note that if $\tilde{w}$ solves (17), then $\tilde{d} = A^*\tilde{w} \in \mathcal{F}_A^*$ from Lemma 2.1, and rearranging the first inclusion in (17) yields $\frac{\langle \tilde{d}, x\rangle}{\|x\|\|\tilde{d}\|} < -t$; therefore $\tilde{d}$ satisfies Condition II of the deep-separation oracle (Definition 2.3). Furthermore, if (17) has no solution, then it is straightforward to show that Condition I of Definition 2.3 is satisfied. This leads to the following approach to constructing a deep-separation oracle for $\mathcal{F}_A$:

given $x \neq 0$, $t > 0$, compute a solution $\tilde{w}$ of (17) or certify that no solution exists. If (17) has no solution, report Condition I and Stop; otherwise (17) has a solution $\tilde{w}$, return $d := A^*\tilde{w}$, report Condition II, and Stop.

Now notice that (17) is a homogeneous conic feasibility problem of the form (13), as it is comprised of the conic inclusion "$(t\|x\|A^*w, \langle w, -Ax\rangle) \in Q^{n+1}$" plus a constraint that the variable w must lie in $S_+^{p\times p}$. Therefore we can apply the results of Section 6.3 if the first inclusion of (17) gives rise to an efficient deep-separation oracle. But indeed the first conic inclusion is a second-order cone inclusion, for which there is an efficient $\alpha = (1/2)$-deep-separation oracle as shown in Section 6.2. Therefore, in the case when (17) has a solution, the results in Sections 6.2 and 6.3 yield a method for doing the requisite computations for Condition II of a (1/2)-deep-separation oracle for $\mathcal{F}_A$.

However, in the case when (17) does not have a solution, it will be necessary to develop a means to certify this infeasibility. To do so, we will run the re-scaled perceptron itself on (17) for a prescribed number of iterations, and assert that (17) does not have a solution, and hence that Condition of I of a deep-separation oracle for $\mathcal{F}_A$ is satisfied, if we do not compute a solution of (17) within this prescribed number of iterations. The following oracle specifies this approach in detail, where the oracle presumes the knowledge of an upper bound L on $\mathcal{C}(A)$:

$(1/2, \delta)$-deep-separation Oracle for $\mathcal{F}_A$ when $K = S_+^{p\times p}$ for $x \neq 0$ and parameter $\hat{t} > 0$, $\delta \in (0, 1]$, and bound $L \geq \mathcal{C}(A)$
Set $t := \hat{t}/2$ and run the re-scaled perceptron algorithm to compute a solution $\tilde{w}$ of (17) for at most $\hat{T} := \max\left\{4096 \ln\left(\frac{1}{\delta}\right), 139n \ln\left(\frac{3L\sqrt{p}}{16n\hat{t}}\right)\right\}$ iterations.
If no solution is computed within $\hat{T}$ iterations, report "assert Condition I is satisfied" and Stop.
If a solution $\tilde{w}$ of (17) is computed, return $d := A^*\tilde{w}$, report Condition II, and Stop.

We will prove:

Theorem 6.1 *Suppose that $K = S_+^{p\times p}$. Then the above oracle is a $(1/2, \delta)$-deep-separation oracle for $\mathcal{F}_A$.*

The oracle described above requires an upper bound L on $\mathcal{C}(A)$. One can use successive doubling of L in conjunction with the iteration bound of Theorem 6.1 to construct a deep-separation oracle for $\mathcal{F}_A$ that does not rely on an upper bound on $\mathcal{C}(A)$. Last of all, as shown in Proposition 7.5 of the Appendix, the width of $\mathcal{F}$ itself can be lower-bounded by Renegar's condition measure:

$$\tau_{\mathcal{F}_A} \geq \frac{\tau_K}{\mathcal{C}(A)} \ . \tag{18}$$

This can be used in combination with the above oracle and successive doubling to construct a $(1/2, \delta)$-deep-separation oracle for $\mathcal{F}_A$ in the case when $K = S_+^{p\times p}$ with an iteration complexity bound that depends polynomially on n, $\ln(p)$, $\ln(\mathcal{C}(A))$, and $\ln(1/\delta)$. Note that the construction of this oracle is less straightforward than in the case when K is composed of half-spaces and/or second-order cones described earlier (Sections 6.1 and 6.2). It is an interesting and open question whether, in the case of $K = S_+^{p\times p}$, a more straightforward and/or more efficient deep-separation oracle for $\mathcal{F}_A$ can be constructed.

Before proving Theorem 6.1, we first analyze the width of the feasibility cone of (17), denoted as $\tilde{\mathcal{F}}_{(t,x)} := \{w : t\|x\|\|A^*w\| + \langle w, Ax\rangle \leq 0,\ w \in K^*\}$. We have:

Proposition 6.1 *For a given $t \in (0, 1/2)$ and $x \neq 0$, suppose that $S_{(t,x)}$ has a solution and let $u \in (0, t)$. Then*

$$\tau_{\tilde{\mathcal{F}}_{(u,x)}} \geq \frac{\tau_{K^*}(t-u)}{3\mathcal{C}(A)} \ ,$$

where $\mathcal{C}(A) = \|A\|_O/\rho(A)$ and $\rho(A)$ is the distance to infeasibility of (16).

Proof. For simplicity we assume with no loss of generality that $\|x\| = 1$ and $\|A\| = 1$. Since $S_{(t,x)}$ has a solution, let $\hat{w}$ satisfy $t\|A^*\hat{w}\| + \langle \hat{w}, Ax\rangle < 0$, $\hat{w} \in$ **int** K^*, and $\|\hat{w}\| = 1$. It follows directly from Theorem 2 of [7] that $\|A^*\hat{w}\| \geq \rho(A)$, where recall that $\rho(A)$ is the distance to infeasibility of A in (16).

Let w° be the center of K^*, whereby $B(w^\circ, \tau_{K^*}) \subset K^*$ and $\|w^\circ\| = 1$. For any d satisfying $\|d\| \le 1$ and any $\beta \ge 0$ it follows that $\hat{w} + \beta w^\circ + \alpha d \in K^*$ so long as $\alpha \le \beta\tau_{K^*}$. Also,

$$\begin{array}{rcl} u\|A^*(\hat{w} + \beta w^\circ + \alpha d)\| + \langle \hat{w} + \beta w^\circ + \alpha d, Ax\rangle & \le & u\|A^*\hat{w}\| + \beta u + \alpha u + \langle \hat{w}, Ax\rangle + \beta + \alpha \\ & \le & (u-t)\|A^*\hat{w}\| + \beta u + \alpha u + \beta + \alpha \\ & \le & (u-t)\rho(A) + \beta u + \alpha u + \beta + \alpha \\ & \le & 0 \end{array}$$

so long as $\alpha \le \hat{\alpha} := \frac{(t-u)\rho(A)}{u+1} - \beta$. Therefore

$$\tau_{\tilde{\mathcal{F}}_{(u,x)}} \ge \frac{\min\left\{\frac{(t-u)\rho(A)}{u+1} - \beta,\ \beta\tau_{K^*}\right\}}{\|\hat{w} + \beta w^\circ\|} \ge \frac{\min\left\{\frac{(t-u)\rho(A)}{u+1} - \beta,\ \beta\tau_{K^*}\right\}}{1+\beta} \ .$$

Let $\beta := \frac{(t-u)\rho(A)}{2(u+1)}$ and substituting in this last expression yields

$$\tau_{\tilde{\mathcal{F}}_{(u,x)}} \ge \frac{(t-u)\rho(A)\tau_{K^*}}{2 + 2u + (t-u)\rho(A)} \ge \frac{(t-u)\rho(A)\tau_{K^*}}{3} = \frac{(t-u)\tau_{K^*}}{3\mathcal{C}(A)}$$

since $\rho(A) \le \|A\| = 1$ and $0 < u \le t \le 1/2$. □

Proof of Theorem 6.1. The oracle attempts to solve $(S_{\hat{t}/2,x})$ where $\hat{t}$ is the relaxation parameter given in the oracle. If the oracle computes a solution $\tilde{w}$ of $(S_{\hat{t}/2,x})$, then it is straightforward to show using Lemma 2.1 that $d := A^*\tilde{w}$ satisfies $d \in \mathcal{F}_A^*$ and $\frac{\langle d,x\rangle}{\|d\|\|x\|} \le -\hat{t}/2$, thus satisfying condition II of Definition 6.2. Next, suppose that it is not true that $\frac{\langle d,x\rangle}{\|d\|\|x\|} \ge -\hat{t}$ for all $d \in \mathbf{ext}\mathcal{F}_A^*$, then $(S_{\hat{t},x})$ has a solution. Set $u = \hat{t}/2$, and define

$$T := \max\left\{4096\ln\left(\frac{1}{\delta}\right), 139n\ln\left(\frac{1}{32n\tau_{\tilde{\mathcal{F}}_{(u,x)}}}\right)\right\} \ .$$

Then Theorem 5.1 states that the probability that the re-scaled perceptron algorithm will fail to compute a solution of $(S_{u,x})$ within T iterations is at most δ. If we can prove that $T \le \hat{T}$, then the probability that algorithm will fail to compute a solution of $(S_{\hat{t}/2,x})$ within $\hat{T}$ iterations is also at most δ, which implies that the oracle will (incorrectly) assert that Condition I is true is at most δ. This will complete the proof.

It therefore remains to show that $T \le \hat{T}$ under the supposition that $(S_{\hat{t},x})$ has a solution. We have $L \ge \mathcal{C}(A)$ and $\tilde{\mathcal{F}}_{(\hat{t},x)} \ne \emptyset$, and it follows from Proposition 6.1 that

$$\frac{1}{32n\tau_{\tilde{\mathcal{F}}_{(u,x)}}} \le \frac{3\mathcal{C}(A)}{32n\tau_{K^*}(\hat{t}-u)} \le \frac{6L}{32n\hat{t}\tau_{K^*}} = \frac{3L\sqrt{p}}{16n\hat{t}}$$

(since $\tau_K^* = \tau_K = \tau_{S_+^{p\times p}} = 1/\sqrt{p}$), whereby $T \le \hat{T}$.□

Acknowledgment. We would like to thank David Gamarnik for discussions leading to a refined proof of Theorem 5.1. The second author acknowledges that this research has been partially supported through the Singapore-MIT Alliance and AFOSR Grant FA9550-08-1-0350. We are also in debt with the referees and associate editor for the quality of their reports, suggestions, and judgement.

7. Appendix We present some basic arithmetic inequalities that are used in the proofs in the body of the paper.

Proposition 7.1 *If $t \in (-1, 2]$, then $\frac{1}{\sqrt{1+t}} \ge 1 - t/2$.*

Proof. For $t \in (-1, 2]$ it follows that $3t^2/4 \ge t^3/4$, whereby $1 \ge 1 - t^2 + t^2/4 + t^3/4 = (1-t/2)^2(1+t)$, and taking square roots and rearranging terms yields the result. □

Proposition 7.2 *If $n \ge 2$, then*

$$\left(1 - \frac{1}{32n}\right)\left(1 - \frac{3}{2048n^2}\right)\left(1 - \frac{1}{8192n^2} + \frac{3}{8n} - \frac{1}{128n^{1.5}}\right) \ge 1 + \frac{1}{3.02n} \ .$$

PROOF. Letting $f(n)$ denote the function on the left side of the inequality to be proved, we have for $n \geq 2$:

$$\begin{aligned} f(n) &\geq \left(1 - \left(\tfrac{1}{32n} + \tfrac{3}{2048n^2}\right)\right)\left(1 + \tfrac{3}{8n} - \left(\tfrac{1}{8192n^2} + \tfrac{1}{128n^{1.5}}\right)\right) \\ &\geq 1 - \left(\tfrac{1}{32n} + \tfrac{3}{2048n^2}\right) + \tfrac{3}{8n} - \left(\tfrac{1}{8192n^2} + \tfrac{1}{128n^{1.5}}\right) - \tfrac{3}{8n}\left(\tfrac{1}{32n} + \tfrac{3}{2048n^2}\right) \\ &\geq 1 - \tfrac{1}{32n} - \tfrac{3}{4096n} + \tfrac{3}{8n} - \tfrac{1}{16384n} - \tfrac{1}{128\sqrt{2}n} - \tfrac{3}{8n}\left(\tfrac{1}{64} + \tfrac{3}{8192}\right) \\ &= 1 + \tfrac{1}{n}\left[\tfrac{3}{8} - \tfrac{1}{32} - \tfrac{3}{4096} - \tfrac{1}{16384} - \tfrac{1}{128\sqrt{2}} - \tfrac{3}{512} - \tfrac{9}{65536}\right] \\ &\geq 1 + \tfrac{1}{3.02n} . \end{aligned}$$

□

PROPOSITION 7.3 *If a, b, c are scalars satisfying $a + b > 0$, $b > 0$, and $c > 0$, then*

$$\left(1 + \frac{a}{b}\right)^c \geq e^{\left(\frac{ac}{a+b}\right)} .$$

PROOF. Using the concavity of the function $\ln(\cdot)$ we have $\ln(1+\alpha) \leq \alpha$ for any $\alpha > -1$. Using this inequality we obtain $c \cdot \ln\left(1 + \frac{a}{b}\right) = -c\ln\left(1 - \frac{a}{a+b}\right) \geq \frac{ac}{a+b}$, and exponentiating yields the result. □

PROPOSITION 7.4 *If $n \geq 2$, then*

$$\left(1 + \frac{1}{3.02n}\right)^{\frac{15}{128}}\left(1 - \frac{1}{32n} - \frac{1}{512n^2}\right)^{\frac{113}{128}} \geq e^{\left(\frac{1}{139n}\right)} .$$

PROOF. Direct substitution establishes the result for $n = 2, 3, 4$, so it remains to show the result for $n \geq 5$. First define the function

$$g(n) := \frac{15n}{1 + 3.02n} - \frac{113n(33/1024)}{n - 33/1024} ,$$

and observe that $g(n)$ is the difference of two terms, the first of which is increasing in n and the second is decreasing in n, therefore $g(n)$ is increasing in n. It then follows for $n \geq 5$ that $g(n) \geq g(5) \geq .99316 \geq 128/139$, where the latter inequality follows from direct substitution. We now prove the result of the proposition. We have for $n \geq 5$:

$$\begin{aligned} \left(1 + \tfrac{1}{3.02n}\right)^{\frac{15}{128}}\left(1 - \tfrac{1}{32n} - \tfrac{1}{512n^2}\right)^{\frac{113}{128}} &\geq \left(1 + \tfrac{1}{3.02n}\right)^{\frac{15}{128}}\left(1 - \tfrac{33}{1024n}\right)^{\frac{113}{128}} \quad \text{(since } n \geq 2\text{)} \\ &\geq e^{\left(\frac{15}{128(1+3.02n)}\right)} e^{-\left(\frac{113(33/1024)}{128(n-33/1024)}\right)} \\ &= e^{\left(\frac{1}{128n}\right)\left[\frac{15n}{1+3.02n} - \frac{113n(33/1024)}{n-33/1024}\right]} \\ &= e^{\left(\frac{1}{128n}\right)[g(n)]} \\ &\geq e^{\left(\frac{1}{128n}\right)\left[\frac{128}{139}\right]} = e^{\left(\frac{1}{139n}\right)} , \end{aligned}$$

where the second inequality uses Proposition 7.3 to bound the first term using $a = 1$, $b = 3.02n$, and $c = 15/128$, and the second term using $a = -33/1024$, $b = n$, and $c = 113/128$. □

PROPOSITION 7.5 *Suppose that (1) has a solution. Then $\tau_{\mathcal{F}_A} \geq \frac{\tau_K}{\mathcal{C}(A)}$.*

PROOF. The proof is an application of Theorem 7 of [7]. Translating to the setting herein, Theorem 7 of [7] states that

$$v^{-1} \geq \rho(A) \tag{19}$$

where

$$v \;:=\; \min_x \;\; \|x\| \qquad \text{s.t.} \;\; Ax - \bar{z} \in K \ ,$$

where $\bar{z} \in K$ is the vector of unit norm that is the "norm approximation vector" of K^*. However, it is shown in the proof of Proposition 2.1 of [6] that $\bar{z}$ is simply the center of the cone K, and hence $B(\bar{z}, \tau_K) \subset K$. Also note from the definition of v that

$$\begin{array}{rcl} v^{-1} & = & \min_x \;\; \theta \\ & & \text{s.t.} \;\; Ax - \theta\bar{z} \in K \\ & & \qquad \|x\| \leq 1 \ . \end{array}$$

Note that any optimal value x in the above optimization problem will satisfy $\|x\| \leq 1$ and $B(x, v^{-1}\tau_K/\|A\|) \subset K$, whereby $\tau_{\mathcal{F}_A} \geq v^{-1}\tau_K/\|A\|$. It then follows that $\tau_{\mathcal{F}_A} \geq v^{-1}\tau_K/\|A\| \geq \rho(A)\tau_K/\|A\| = \tau_K/\mathcal{C}(A)$. □

7.1 Deep-separation oracle with approximate eigenvalues and eigenvectors In this Appendix we analyze the deep-separation for $K = S^{p\times p}_+$ when one can only approximate the minimum eigenvalue and associated eigenvector. Let x be the current iterate such that

$$B_k x \notin K$$

and let $\lambda = \langle v, B_k x v\rangle = \langle vv^T, B_k x\rangle < 0$ denote the minimum eigenvalue of $B_k x$ and v a corresponding eigenvector. Without loss of generality we can assume $\|x\| = 1$ (note that λ scales linearly with $\|x\|$). Let $\hat{\lambda}$ and $\hat{v}$ denote the numerical approximations for λ and v.

We will divide the analysis in two cases: $|\lambda|$ small, and $|\lambda|$ large. (This is needed since it might be hard to approximate $|\lambda|$ numerically when it is arbitrary small.)

First we consider the case that $|\lambda|$ is "small", namely we assume that

$$B_k x \succcurlyeq -\mu B_k B_k^* I$$

where $I \in S^{p\times p}_+$ denotes the appropriate identity matrix, and $\mu := \frac{\sigma^2}{8}\frac{\|x\|}{\|B_k^* I\|}$ with $\sigma = 1/(32n)$. (Note that this can be tested efficiently, see discussion below.) In this case it is easy to add a positive component to all directions without affecting the potential function $\langle x^+, z^k\rangle/\|x^+\|$ too much. More precisely, the updated iterate

$$x^+ \leftarrow x + \mu B_k^* I$$

satisfies

$$\langle x^+, z^k\rangle = \langle x, z^k\rangle + \mu\langle B_k^* I, z^k\rangle = \langle x, z^k\rangle + \mu\langle I, B_k z^k\rangle \geq \langle x, z^k\rangle + \mu\|B_k z^k\|/\sqrt{n}$$

since $B_k z^k \in S^{p\times p}$, and

$$\|x^+\|^2 = \|x\|^2 + 2\mu\langle x, B_k^* I\rangle + \mu^2\|B_k^* I\|^2 \leq \|x\|^2\left(1 + \frac{\sigma^2}{8}\right)^2 .$$

Note that this does not affect the bound on $\tau_{\mathcal{F}_{\hat{A}}}$ obtained in (i) on Lemma 5.2. On the other hand for (ii), with probability at least $1/8$ we have that (as defined in Lemma 5.2) $\langle \bar{z}, \bar{x}\rangle \geq \frac{1}{\sqrt{n}}\left(1 - \frac{\sigma^2}{8}\right)$ so that

$$\begin{array}{rcl} \|\hat{z}\|^2 & \leq & 1 - \frac{3}{4}\left(\frac{1}{\sqrt{n}} - \frac{\sigma^2}{8n^{1/2}}\right)^2 + \frac{\tau}{2}\left(\frac{1}{\sqrt{n}} - \frac{\sigma^2}{8n^{1/2}}\right) + \frac{\tau^2}{4} \\ & = & 1 - \frac{3}{4n} + \frac{\tau}{2\sqrt{n}} + \frac{\tau^2}{4} - \frac{\tau\sigma^2}{16n^{1/2}} - \frac{3\sigma^4}{256n} + \frac{3\sigma^2}{16n}. \end{array}$$

The bound on Proposition 7.2 needs to be slightly modified (assuming $n \geq 10$) to

$$\begin{aligned}
f(n) &\geq \left(1-\left(\tfrac{1}{32n}+\tfrac{3}{2048n^2}\right)\right)\left(1+\tfrac{3}{8n}-\left(\tfrac{1}{8192n^2}+\tfrac{1}{128n^{1.5}}+\tfrac{3\sigma^2}{32n}\right)\right)\\
&\geq 1-\left(\tfrac{1}{32n}+\tfrac{3}{2048n^2}\right)+\tfrac{3}{8n}-\left(\tfrac{1}{8192n^2}+\tfrac{1}{128n^{1.5}}+\tfrac{3\sigma^2}{32n}\right)-\tfrac{3}{8n}\left(\tfrac{1}{32n}+\tfrac{3}{2048n^2}\right)\\
&\geq 1-\tfrac{1}{32n}-\tfrac{3}{20480n}+\tfrac{3}{8n}-\tfrac{1}{81920n}-\tfrac{1}{128\sqrt{10}n}-\tfrac{3}{3200n}-\tfrac{3}{8n}\left(\tfrac{1}{320}+\tfrac{3}{20480}\right)\\
&= 1+\tfrac{1}{n}\left[\tfrac{3}{8}-\tfrac{1}{32}-\tfrac{3}{20480}-\tfrac{1}{81920}-\tfrac{1}{128\sqrt{10}}-\tfrac{3}{3200}-\tfrac{3}{2560}-\tfrac{9}{163840}\right]\\
&\geq 1+\tfrac{1}{2.958n}\geq 1+\tfrac{1}{3.02n}\ .
\end{aligned}$$

Therefore the analysis of the outer number of iterations does not change by this truncation. However, the number of iterations on Step 4 (Perceptron Improvement Phase) needs to account for this decrease in the potential. The truncation makes all the eigenvalues non-negative. Therefore either we terminate Step 4 or the deep-separation oracle returns a direction $d \in \mathcal{F}_A^*$. In this case the potential function is improved by a factor of $\sqrt{1-\sigma^2}$. At this point one could possibly call the truncation procedure again. This implies that every two iterations the improvement in the potential would be of at least

$$\frac{1}{\sqrt{1-\sigma^2}\left(1+\frac{\sigma^2}{8}\right)} \geq \left(1+\frac{\sigma^2}{2}\right)\left(1-\frac{\sigma^2}{8}\right) \geq 1+\frac{\sigma^2}{4}.$$

Therefore it suffices to alter the number of loops within Step 4(a) to be at most $\lfloor(2/\sigma^2)\ln(n)\rfloor$.

Next assume that $|\lambda|$ is large, namely

$$B_k x \not\succeq -\mu B_k B_k^* I$$

which implies that

$$\lambda \leq -\mu\left\langle v, B_k B_k^* I v\right\rangle = -\mu\left\langle B_k^* vv^T, B_k^* I\right\rangle \leq \frac{-\mu}{\sqrt{n}}$$

since $B_k B_k^*(vv^T) \in S^{p\times p}$ by Lemma 6.1 and $B_k B_k^*(vv^T) \succcurlyeq vv^T$.

In this case we compute $\hat\lambda$ and $\hat v$, and set $\hat d = (\hat v \hat v^T) \in S_+^{p\times p}$. Next define the next iterate as

$$x^+ \leftarrow x + 2\theta B_k^* d \qquad (20)$$

where $\theta := -\left\langle \hat d, B_k x\right\rangle/\|B_k^*\hat d\|^2 = -\hat\lambda/\|B_k^*\hat d\|^2$. It then follows that

$$\left\langle x^+, z^k\right\rangle = \left\langle x, z^k\right\rangle + 2\theta\left\langle B_k^*\hat d, z^k\right\rangle = \left\langle x, z^k\right\rangle + 2\theta\left\langle \hat d, B_k z^k\right\rangle \geq \left\langle x, z^k\right\rangle$$

since $B_k z^k \in K$ and hence $\left\langle \hat d, B_k z^k\right\rangle \geq 0$. Furthermore, from the particular choice of θ we have

$$\|x^+\|^2 = \|x\|^2 + 4\theta\left\langle x, B_k^*\hat d\right\rangle + 4\theta^2\|B_k^*\hat d\|^2 = \|x\|^2\ ,$$

and hence the potential function $\left\langle z^k, x\right\rangle/\|x\|$ is non-decreasing if we replace x by x^+.

As before, if all previous iterates satisfied $B_i x_{i+1} \in K$, $i = 1, \ldots, k-1$, then we have from Lemma 6.1 (using $k-1$ instead of k) that

$$B_k x^+ - B_k x = \theta B_k B_k^* \hat d \in K$$

and furthermore from the choice of θ we have

$$\begin{aligned}
v^T(B_k x^+)v &= v^T(B_k x + 2\theta B_k B_k^*\hat d)v = \left\langle B_k x + 2\theta B_k B_k^*\hat d, vv^T\right\rangle\\
&= \left\langle B_k x, vv^T\right\rangle + 2\theta\left\langle B_k^*\hat v\hat v^T, B_k^* vv^T\right\rangle\\
&= \lambda - 2\hat\lambda\left\langle \tfrac{B_k^*\hat v\hat v^T}{\|B_k^*\hat v\hat v^T\|}, \tfrac{B_k^* vv^T}{\|B_k^*\hat v\hat v^T\|}\right\rangle\\
&= \lambda - 2\hat\lambda\left(1-\left\langle \tfrac{B_k^*\hat v\hat v^T}{\|B_k^*\hat v\hat v^T\|}, \tfrac{B_k^*(vv^T-\hat v\hat v^T)}{\|B_k^*\hat v\hat v^T\|}\right\rangle\right) \geq -\lambda/8
\end{aligned}$$

if we ensure that $\left\langle \frac{B_k^*\hat v\hat v}{\|B_k^*\hat v\hat v\|}, \frac{B_k^*(vv^T-\hat v\hat v^T)}{\|B_k^*\hat v\hat v^T\|}\right\rangle \leq \frac{1}{4}$ and $|\hat\lambda-\lambda| \leq \frac{|\lambda|}{4}$ where $\lambda \geq \frac{\mu}{4\sqrt{n}}$. Therefore the same argument described in the end of Section 6.3 still applies.

The last issue is the computation complexity of $\hat{\lambda}$ and $\hat{v}$ within the needed precision. Note that $B_k^* \hat{v}\hat{v}^T \succcurlyeq \hat{v}\hat{v}^T$ so that $\|B_k^*\hat{v}\hat{v}^T\| \geq \|\hat{v}\| = 1$, $\|B_k^*\| \leq 2^k$, and $\mu \geq \frac{1}{2^{k+12}n^3}$. Therefore a simple bound on the needed precisions are

$$\|v - \hat{v}\| \leq \frac{1}{2^{k+2}} \text{ and } |\lambda - \hat{\lambda}| \leq \frac{1}{2^{k+14}n^3}.$$

Since k is bounded by T and the computational complexity of approximating v and λ is logarithmic on the desired precision, the oracle can be implemented in polynomial time (see Ye [23], Renegar [16], Vavasis and Zippel [21], and Fu, Luo and Ye [8] for the complexity result).

References

[1] A. Belloni and R. M. Freund, *On the second-order feasibility cone: primal-dual representation and efficient projection*, SIAM Journal on Optimization **19** (2008), no. 3, 1073–1092.

[2] A. Berman, *Cones, matrices, and mathematical programming*, Springer-Verlag, New York, 1973.

[3] D. Bertsimas and S. Vempala, *Solving convex programs by random walks*, Journal of the ACM **51** (2004), no. 4, 540–556.

[4] A. Blum, A. Frieze, R. Kannan, and S. Vempala, *A polynomial-time algorithm for learning noisy linear threashold functions*, Algorithmica **22** (1998), no. 1, 35–52.

[5] J. Dunagan and S. Vempala, *A simple polynomial-time rescaling algorithm for solving linear programs*, Mathematical Programming (2007), no. DOI 10.1007/s10107-007-0095-7.

[6] R. M. Freund and J. R. Vera, *Condition-based complexity of convex optimization in conic linear form via the ellipsoid algorithm*, SIAM Journal on Optimization **10** (1999), no. 1, 155–176.

[7] ______, *Some characterizations and properties of the "distance to ill-posedness" and the condition measure of a conic linear system*, Mathematical Programming **86** (1999), no. 2, 225–260.

[8] M. Fu, Z.-Q. Luo, and Y. Ye, *Approximation algorithms for quadratic programming*, Journal of Combinatorial Optimization **2** (1998).

[9] I. S. Gradshtein and I. M. Ryzhik, *Table of integrals, series, and product*, Academic Press, 2007.

[10] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combiantorial optimization*, second ed., Springer-Verlag, Berlin, 1994.

[11] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica **4** (1984), no. 4, 373–395.

[12] L.G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Math. Dokl. **20** (1979), no. 1, 191–194.

[13] R. Motwani and P. Raghanav, *Randomized algorithms*, Cambridge University Press, 1995.

[14] Y. Nesterov and A Nemirovskii, *Interior-point polynomial algorithms in convex programming*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1993.

[15] G. Pataki, *On the closedness of the linear image of a closed convex cone*, Technical Report, University of North Carolina, TR-02-3 Department of Operations Research 1992.

[16] J. Renegar, *On the worst-case arithmetic complexity of approximating zeros of polynomials*, Journal of Complexity **3** (1987).

[17] J. Renegar, *Some perturbation theory for linear programming*, Mathematical Programming **65** (1994), no. 1, 73–91.

[18] ______, *Linear programming, complexity theory, and elementary functional analysis*, Mathematical Programming **70** (1995), no. 3, 279–351.

[19] R. T. Rockafellar, *Convex analysis*, Princeton University Press, Princeton, New Jersey, 1970.

[20] F. Rosenblatt, *Principles of neurodynamics*, Spartan Books, Washington, DC, 1962.

[21] S.A. Vavasis and R. Zippel, *Proving polynomial time for sphere-constrained quadratic programming*, Department of Computer Science, Cornell University, Ithaca, NY, Technical Report (1990).

[22] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of semidefinite programming*, Kluwer Academic Publishers, 2000.

[23] Y. Ye, *On affine scaling algorithms for nonconvex quadratic programming*, Mathematical Programming **56** (1992).