Polynomial Cutting Plane Algorithms for Two-Stage Stochastic Linear Programs Based on Ellipsoids, Volumetric Centers and Analytic Centers

K. A. Ariyawansa [a,*,1],

[a] *Department of Mathematics, Washington State University, Pullman, WA 99164-3113, USA.*

P. L. Jiang [b,2]

[b] *Professional Services, Delta Dental Plan of Minnesotta, 3560 Delta Dental Drive, eagan, MN 55122-3166, USA.*

Abstract

Traditional simplex-based algorithms for two-stage stochastic linear programs can be broadly divided into two groups: (a) those that *explicitly* exploit the structure of the equivalent large-scale linear program and (b) those based on cutting planes (or equivalently on decomposition) that *implicitly* exploit that structure. Algorithms of group (b) are in general preferred. In 1988, following the work of Karmarkar for general linear programs, Birge and Qi [11] proposed a specialization of Karmarkar's algorithm for two-stage stochastic linear programs. The algorithm of Birge and Qi [11] is the first interior point analog of group (a). Several other authors have studied related and different interior point analogs of group (a). Birge and Qi [11] also presented an analysis of the computational complexity of their algorithm. This analysis indicates that the computational complexity (in terms of total arithmetic operations) of their algorithm is in general smaller than that of the Karmarkar's algorithm (applied without modification), and is *quadratic* in the number of realizations. At the time the present paper was initially prepared in 1996, the only work on interior point analogs of the preferred group (b) available was the paper by by Bahn, du Merle, Goffin and Vial [6], who presented an algorithm based on analytic centers. However, they did not present results on the complexity of their algorithm. In this paper, we present three classes of interior point analogs of group (b) based respectively on the ellipsoid algorithm (Khachiyan [29]), on the notion of volumetric center (Vaidya [38]) and on the notion of analytic center (Sonnevend [35]). We also present complexity results which indicate that the complexities (in terms of total arithmetic operations) of certain members of our three classes of algorithms are *linear* in the number of realizations.

Key words: Stochastic programming, cutting planes, Van Slyke and Wets algorithm, ellipsoid algorithm, analytic center, volumetric center.

1 Introduction

Consider the problem

$$\begin{array}{rl} \text{minimize} & Z(x) := c^{\mathsf{T}}x + \mathcal{Q}(x) \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array} \tag{1}$$

where[3] $x \in \mathbb{R}^{n_1}$ is the decision variable, and $A \in \mathbb{R}^{m_1 \times n_1}$, $b \in \mathbb{R}^{m_1}$ and $c \in \mathbb{R}^{n_1}$ are deterministic data, and $\mathcal{Q}(x) := E[Q(x, \boldsymbol{q}, \boldsymbol{h}, \boldsymbol{M}, \boldsymbol{T})]$, the expectation with respect to random data $\boldsymbol{q} \in \mathbb{R}^{n_1}$, $\boldsymbol{h} \in \mathbb{R}^{m_2}$, $\boldsymbol{M} \in \mathbb{R}^{m_2 \times n_2}$, $\boldsymbol{T} \in \mathbb{R}^{m_2 \times n_1}$ of the value of a function Q. The dependence of function Q on x and a realization q, h, M and T of $\boldsymbol{q}$, $\boldsymbol{h}$, $\boldsymbol{M}$ and $\boldsymbol{T}$ respectively is specified by

$$Q(x, q, h, M, T) := \inf_{y \in \mathbb{R}^{n_2}} \{q^{\mathsf{T}}y : My = h - Tx, y \geq 0\}. \tag{2}$$

We assume that random data have the given discrete probability distribution

$$\{((q^l, h^l, M^l, T^l), p^l), l = 1, 2, \ldots, K\} \tag{3}$$

so that $\mathcal{Q}(x) = \sum_{l=1}^{K} p^l Q(x, q^l, h^l, M^l, T^l)$. In the rest of the paper we use the term *two-stage stochastic program with recourse* to refer to the problem defined by (1,2,3) as outlined above.

Two-stage stochastic programs arise from decision making models that fit into the following generic context. The vector x is a decision that has to be made *at*

[*] Corresponding author.
Email addresses: `ari@math.wsu.edu` (K. A. Ariyawansa), `pjiang@deltadental.org` (P. L. Jiang).
URL: `www.wsu.edu:8080/ ari` (K. A. Ariyawansa).

[1] Research of this author was supported in part by NSF Grant CCR-9403892 and by the U.S. Department of Army under grant DAAD 19-00-1-0465.

[2] Supported in part by funds provided by Washington State University in the form of a summer research assistantship.

[3] We use the symbol ':=' to indicate equality by definition in mathematical contexts, and assignment in statement of algorithms. Bold-face letters denote random variables while corresponding normal-face letters denote their realizations. We only use column vectors and use the superscript 'T' to denote transposition.

present (the stage 1), and after a realization q^l, h^l, M^l and T^l of the random data becomes available at a *later time* (the stage 2), a recourse decision y^l may be taken if $h^l - T^l x \neq 0$ with an associated technological matrix M^l and unit cost vector q^l.

"Usual" instances of stochastic programs have the following three features due to the above decision making models from which they arise. First, the number of realizations K is "large". Second, for "most" x satisfying $Ax = b$ and $x \geq 0$, there exists $y^l \geq 0$ such that $M^l y^l = h^l - T^l x$ for $l = 1, 2, \ldots, K$. Third, the decision variable x is more important than the decision variables y^l, $l = 1, 2, \ldots, K$.

We refer to the representation (1,2,3) of two-stage stochastic linear programs as the *recourse formulation.* The representation (1,2,3) is mathematically equivalent to the linear program (lp)

$$
\begin{array}{lrcccl}
\text{minimize} & Z = & c^{\mathsf{T}} x & + & \sum_{l=1}^{K} p^l (q^l)^{\mathsf{T}} y^l & \\
\text{subject to} & & Ax & & & = b \\
 & & T^l x & + & M^l y^l & = h^l,\ l = 1, 2, \ldots, K \\
 & & x, & & y^l & \geq 0,\ l = 1, 2, \ldots, K
\end{array}
\tag{4}
$$

which we refer as the *full-lp formulation.* Note that in the recourse formulation the decision variable x is treated directly receiving prominence over the decision variables y^l, $l = 1, 2, \ldots, K$, whereas in the full-lp formulation, decision variables x and y^l, $l = 1, 2, \ldots, K$ receive similar prominence. Note also that formulation (4) has a "large" number of constraints, and even though "most" of them are "loose" they are explicitly included in (4). On the other hand, these constraints are taken into account implicitly in the recourse-formulation. Therefore, given our comments above on "usual" instances of stochastic programs, the recourse formulation is more natural than the full-lp formulation.

Algorithms for two-stage stochastic programs can be broadly divided into two groups depending on whether the recourse formulation (1,2,3), or the full-lp formulation (4) is emphasized. Simplex-based algorithms due to Kall [28], Straziky [36], and Wets [41] emphasize the full-lp formulation and explicitly exploit the structure of the lp in (4) to reduce the computational work. These algorithms solve the dual of (4) and utilize its structure to work with a basis requiring $n_1^2 + Kn_2^2$ locations rather than one required with $(n_1 + Kn_2)^2$ locations if that structure were ignored.

Birge and Qi [11] presented the first interior point algorithm of this type. Let

$$\hat{A} := \begin{bmatrix} A & 0 & 0 & \dots & 0 \\ T^1 & M^1 & 0 & \dots & 0 \\ T^2 & 0 & M^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T^K & 0 & 0 & \dots & M^K \end{bmatrix}$$

be the constraint matrix of the lp (4), and $\hat{b} := [(b)^\mathsf{T}, (h^1)^\mathsf{T}, (h^2)^\mathsf{T}, \dots, (h^K)^\mathsf{T}]^\mathsf{T}$ be its right-hand-side. Birge and Qi [11] showed how the structure of $\hat{A}$ can be utilized to solve the system $(\hat{A}D^2\hat{A}^\mathsf{T})d = \hat{b}$ arising in interior point methods for the lp (4), so that the complexity of the resulting specialization of Karmarkar's algorithm is

$$O((n^{0.5}n_2^2 + n\max\{n_1, n_2\} + n_1^3)nL) \tag{5}$$

arithmetic operations, where $n := n_1 + Kn_2$. This is in general smaller than $O(n^{3.5}L)$ complexity of Karmarkar's algorithm applied directly on the lp (4). Note that the complexity bound (5) is quadratic[4] in K. Birge and Holmes [10], and Jessup, Yang and Zenios [26] present efficient implementations of the Birge and Qi [11] algorithm. The "split variable technique" presented by Lustig, Mulvey and Carpenter [30] (see also [14]) for formulating two-stage stochastic programs for interior point methods also works with the full-lp formulation.

The most popular simplex-based algorithm based on the recourse formulation is the one originally due to Van Slyke and Wets [40] and later extended by Wets [41]. (See also Dantzig and Madansky [16].) The operations of the algorithm can be interpreted as being based on *cutting planes generated from the minimizer of an lp relaxation* of the problem. It terminates after a finite number of iterations, but there are no polynomial complexity results. See [9,22,3] for implementations of the algorithm.

As indicated above, the recourse formulation is more natural than the full-lp formulation. Therefore, given the rapid spread of research on interior point methods to areas of mathematical programming other than linear programming, one would expect to have interior point methods for stochastic linear programs based on the recourse formulation. However, the only *direct* work

[4] It is possible to derive an algorithm based on full-lp formulation similar to the algorithm of Birge and Qi [11] but specializing the path-following algorithm rather than Karmarkar's algorithm with a complexity that depends on $K^{1.5}$ as opposed to K^2. This follows from the work of Renegar [34] for general linear programming. The work of Birge and Qi [11] was performed prior to that of Renegar [34].

known to us at the time the present paper was initially[5] prepared was due to Bahn, du Merle, Goffin and Vial [6].

The operations of the algorithm of Bahn et al. [6] may be interpreted as being based on *cutting planes generated from the analytic center [35] of a certain set of localization.* Bahn et al. [6] give no complexity results for their algorithm. Their computational results indicate that the algorithm compares well with the simplex-based cutting plane methods.

Although there is no work other than that of Bahn et al. [6] directly related to interior point algorithms for stochastic linear programs based on the recourse formulation, papers [37,38,23,42,24,43,17,31,20,5,2,4,33] are indirectly related since they deal with topics such as cutting planes, decomposition and column generation in connection with interior point methods. In particular, our work is motivated by the elegant papers by Vaidya [37–39], Atkinson and Vaidya [4], and by Anstreicher [2,1]. These papers mainly deal with the *convex feasibility* problem: given convex $S \subset \mathbb{R}^n$ find $x \in S$. There are several important features of this work that motivated us to adapt it for obtaining algorithms for stochastic linear programs based on the recourse formulation. First, the algorithms do not explicitly use a full representation of S. Instead, they assume that there is an oracle with the following properties. It takes any $\bar{x} \in \mathbb{R}^n$ as input, and if $\bar{x} \in S$ it indicates so. If $\bar{x} \notin S$ it returns a $D \in \mathbb{R}^n$ and $d \in \mathbb{R}$ such that for all $x \in S$, $D^{\mathsf{T}}x \geq d$ and $D^{\mathsf{T}}\bar{x} < d$; i.e. D and d specify a cutting plane. Second, while in some of the algorithms these cutting planes are generated based on the notion of analytic center of a polyhedral set, in some other algorithms they are generated based on the newer notion of volumetric center due to Vaidya. Finally, these authors relate their work to the ellipsoid method [29,12] which despite its poor practical performance in some problems is theoretically important. Indeed, the work of Vaidya [37,38], Atkinson and Vaidya [4], and Anstreicher [2,1] is such that an ideal application is in obtaining “interior point cutting plane algorithms” for stochastic linear programs based on the recourse formulation with proofs of polynomial complexity results.

In this paper we derive three classes of algorithms for two-stage stochastic linear programs based on the recourse formulation (1,2,3). The three classes utilize the ellipsoid method [29,12], the notion of volumetric center [38], and

[5] This paper was initially prepared and submitted for publication in 1996. It is part of the doctoral dissertation [27] of the second author completed in 1997. While this paper was being refereed and revised, a few papers containing material relevant to this paper (some referencing the initial version of this paper and [27]) became available. The results in this paper however are still new. The present version therefore contains a new §7 in which we indicate the connections of these contributions to our work. However, references to literature in other sections of the paper have not been altered.

the notion of analytic center [35] respectively. We also obtain polynomial complexity results for the three classes of algorithms. These results indicate that all three classes contain members with complexities that are linear in K, in comparison to (5) for the algorithm of Birge and Qi [11] that depends quadratically on K. Before proceeding further, it is worthwhile pointing out that the realtionship of our work to the body of work in [37,38,23,42,24,43,17,31,20,5,2,4,33] is similar to the relatiosnhip of the work of Van Slyke and Wets [40] to the work of Benders [7] (and Dantzig and Wolfe [15]). See also the interesting footnote in [40, p. 639] on a comment by E. Balas on that latter relationship.

The rest of this paper is structured as follows. We first consider the case of (1,2,3) with $K := 1$:

$$\begin{array}{rl} \text{minimize} & Z(x) := c^{\mathsf{T}}x + \mathcal{Q}(x) \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array} \tag{6}$$

with

$$\mathcal{Q}(x) := \inf_{y \in \mathbb{R}^{n_2}} \{q^{\mathsf{T}}y : My = h - Tx, y \geq 0\} \tag{7}$$

where we have omitted the superscript 1. As we shall see in §6, we do not lose generality, and our presentation and notation become considerably easy. The corresponding full-lp formulation is

$$\begin{array}{rrcccl} \text{minimize} & Z = & c^{\mathsf{T}}x & + & q^{\mathsf{T}}y & \\ \text{subject to} & & Ax & & & = b \\ & & Tx & + & My & = h \\ & & x, & & y & \geq 0 \end{array} \tag{8}$$

which is referred to as an L-shaped lp by Van Slyke and Wets [40]. Note that in (8) and consequently in (6,7) we can assume without loss of generality that

$b = 0$ since (8) is equivalent to

$$\begin{array}{llcccc}
\text{minimize} & Z = [c^{\mathsf{T}}, 0] \begin{bmatrix} x \\ s \end{bmatrix} & + & q^{\mathsf{T}} y & & \\
\text{subject to} & [A, -b] \begin{bmatrix} x \\ s \end{bmatrix} & & & = & 0 \\
& \begin{bmatrix} T & 0 \\ 0^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} & + & \begin{bmatrix} M \\ 0^{\mathsf{T}} \end{bmatrix} y & = & \begin{bmatrix} h \\ 1 \end{bmatrix} \\
& \begin{bmatrix} x \\ s \end{bmatrix}, & & y & \geq & 0.
\end{array}$$

In fact, we can similarly assume that $b = 0$ in (4) and (1,2,3) without loss of generality, and we shall do so in the rest of the paper.

Let

$$\begin{aligned}
&S_1 := \{x \in \mathbb{R}^{n_1} : Ax = 0\}, \\
&S_2 := \{x \in \mathbb{R}^{n_1} : x \geq 0\}, \text{ and} \\
&S_3 := \{x \in \mathbb{R}^{n_1} : (\exists y \in \mathbb{R}^{n_2} : My = h - Tx, y \geq 0)\}.
\end{aligned}$$

Then (6) is equivalent to

$$\begin{array}{rl}
\text{minimize} & Z(x) := c^{\mathsf{T}} x + \theta \\
\text{subject to} & \mathcal{Q}(x) \leq \theta \\
& x \in S := S_1 \cap S_2 \cap S_3
\end{array} \tag{9}$$

where $\mathcal{Q}$ is as defined in (7). Throughout the paper we assume that

(A1) A has full row rank, and
(A2) (9) has a minimizer $[(x^*)^{\mathsf{T}}, \theta^*]^{\mathsf{T}}$.

In §§2, 3 and 4 we present three classes of algorithms for (9) based respectively on the ellipsoid method, on volumetric centers, and on analytic centers together with results on their complexities. In doing so, we need to frequently use projections $\bar{z}$ of $z := [x^{\mathsf{T}}, \theta]^{\mathsf{T}} \in \mathbb{R}^{n_1+1}$ onto $S_1' := \{z : z = [x^{\mathsf{T}}, \theta]^{\mathsf{T}}, x \in S_1\}$, the $(n_1 - m_1 + 1)$-dimensional “vertical” hyperplane corresponding to S_1 in $\mathbb{R}^{n_1+1}$. In fact, we have $\bar{z} := [\bar{x}^\mathsf{T}, \bar{\theta}]^\mathsf{T} = Pz$ where

$$P = \begin{bmatrix} P_A & 0 \\ 0^\mathsf{T} & 1 \end{bmatrix}, \quad \text{and} \quad P_A := I - A^\mathsf{T}(AA^\mathsf{T})^{-1}A.$$

(Note that $\bar{\theta} = \theta$.) In the rest of the paper a “bar” over vectors (and their components) in $\mathbb{R}^{n_1+1}$ indicates such projections (and their components). In these three sections we assume that we have an oracle that does the following. Suppose that the oracle is presented with $[\bar{x}^\mathsf{T}, \bar{\theta}]^\mathsf{T}$ satisfying $A\bar{x} = 0$ and $\bar{x} \geq 0$ (i.e. $\bar{x} \in S_1 \cap S_2$). Then the oracle decides whether $\bar{x} \in S_3$ or not. If it decides that $\bar{x} \notin S_3$, it returns $D \in \mathbb{R}^{n_1}$ and $d \in \mathbb{R}$ such that $D^\mathsf{T} x \geq d$ for all $x \in S_3$ and $D^\mathsf{T}\bar{x} < d$. If the oracle concludes that $\bar{x} \in S_3$, then it decides whether $\mathcal{Q}(\bar{x}) \leq \bar{\theta}$ (i.e. whether $[\bar{x}^\mathsf{T}, \bar{\theta}]^\mathsf{T}$ is feasible for (9)) or not. If it decides that $\mathcal{Q}(\bar{x}) > \bar{\theta}$ then it returns $E \in \mathbb{R}^{n_1}$ and $e \in \mathbb{R}$ such that $E^\mathsf{T} x + \theta \geq e$ for all $[x^\mathsf{T}, \theta]^\mathsf{T}$ feasible for (9) and $E^\mathsf{T}\bar{x} + \bar{\theta} < e$. If it decides that $\mathcal{Q}(\bar{x}) \leq \bar{\theta}$ then it returns θ', the value of the objective function in the lp (7) at any feasible point, that satisfies $\theta' \leq \bar{\theta}$. Readers familiar with the work of Van Slyke and Wets [40] on simplex-based cutting plane algorithms for stochastic linear programs would recognize that these are the analogs of the “feasibility cuts” and “optimality cuts” in the terminology of [40]. However, as we shall see in §5, the oracle does not necessarily have to solve lp’s exactly to specify these cuts. We shall indicate several ways the work of the oracle may be performed in §5 together with estimates of the corresponding costs. In §6 we shall indicate how the algorithms stated in §§2,3 and 4 for the special case with $K := 1$ can be used to obtain corresponding algorithms for (1,2,3). We conclude the paper in §7, where we discuss the relationship of the contribution of this paper to those of several other papers that have appeared after this paper was initially submitted for publication and was in the process of being refereed and revised.

2 A class of cutting plane algorithms for L-shaped linear programs based on the ellipsoid method

The theoretical significance of the ellipsoid method [29,12] in linear, nonlinear and combinatorial optimization is profound. See [25] for example. To the best of our knowledge however, there is no ellipsoidal analog of the simplex-based cutting plane algorithms such as that of Van Slyke and Wets [40] for L-shaped lp’s (9) and stochastic linear programs (1,2,3). In this section, we present a class of algorithms for L-shaped lp’s based on the ellipsoid method.

We begin with $x_0 \in \mathbb{R}^{n_1}$, $\theta_0 \in \mathbb{R}$ and an ellipsoid $\mathcal{E}_0 := \{z \in \mathbb{R}^{n_1+1} : (z - z_0)^{\mathsf{T}} B_0^{-1}(z - z_0) \le 1\}$ where $z_0 := [x_0^{\mathsf{T}}, \theta_0]^{\mathsf{T}}$ and $B_0 \in \mathbb{R}^{(n_1+1)\times(n_1+1)}$ is symmetric and positive definite, so that $\mathrm{int}\,(\mathcal{E}_0)$ contains a minimizer $z^* := [(x^*)^{\mathsf{T}}, \theta^*]^{\mathsf{T}}$ of (9). We can choose $B_0 := 2^{2L} I$ with $L > 0$ sufficiently large. At the beginning of iteration $k(\ge 0)$ we have the ellipsoid $\mathcal{E}_k := \{z \in \mathbb{R}^{n_1+1} : (z - z_k)^{\mathsf{T}} B_k^{-1}(z - z_k) \le 1\}$ with center z_k and symmetric, positive definite $B_k \in \mathbb{R}^{(n_1+1)\times(n_1+1)}$ with $z^* \in \mathrm{int}\,(\mathcal{E}_k)$. Now we compute the projection $\bar{z}_k := [\bar{x}_k^{\mathsf{T}}, \bar{\theta}_k]^{\mathsf{T}} = P z_k$ so that $\bar{x}_k = P_A x_k \in S_1$ and $\bar{\theta}_k = \theta_k$.

Case 1. $\bar{x}_k \notin S_2$.

Then there exists j_k with $1 \le j_k \le n_1$ such that $e_{j_k}^{\mathsf{T}} \bar{x}_k < 0$. We add the cut

$$\bar{e}_{j_k}^{\mathsf{T}} x \ge 0 \tag{10}$$

where $\bar{e}_{j_k} := P_A e_{j_k}$. All x such that $\bar{x} \in S_1 \cap S_2$ satisfy the cut (10) (see Lemma 1 below) while x_k violates it.

Case 2. $\bar{x}_k \in S_2$.

So, $A\bar{x}_k = 0$ and $\bar{x}_k \ge 0$ and we call the oracle.

Subcase 2.1. The oracle indicates that $\bar{x}_k \notin S_3$.

In this case as we indicated in §1, the oracle returns $D_k \in \mathbb{R}^{n_1}$ and $d_k \in \mathbb{R}$ such that $D_k^{\mathsf{T}} x \ge d_k$ for all $x \in S_3$ and $D_k^{\mathsf{T}} \bar{x}_k < d_k$. We add the cut

$$\bar{D}_k^{\mathsf{T}} x \ge d_k \tag{11}$$

where $\bar{D}_k := P_A D_k$. The cut (11) is satisfied by any x such that $\bar{x} \in S_1 \cap S_2 \cap S_3$ (see Lemma 1 below) and is violated by $\bar{x}_k$.

Subcase 2.2. The oracle indicates that $\bar{x}_k \in S_3$.

In this case, as indicated in §1, the oracle decides whether $\mathcal{Q}(\bar{x}_k) \le \bar{\theta}_k$ or not. We have two further subcases.

Subcase 2.2.1. The oracle decides $\mathcal{Q}(\bar{x}_k) > \bar{\theta}_k$.

In this case $[\bar{x}_k^{\mathsf{T}}, \bar{\theta}_k]^{\mathsf{T}}$ is not feasible for (9) and the oracle returns $E_k \in \mathbb{R}^{n_1}$ and $e_k \in \mathbb{R}$ such that $E_k^{\mathsf{T}} x + \theta \ge e_k$ for all $[x^{\mathsf{T}}, \theta]^{\mathsf{T}}$ feasible for (9) and $E_k^{\mathsf{T}} \bar{x}_k + \bar{\theta}_k < e_k$. We add the cut

$$\bar{E}_k^{\mathsf{T}} x + \theta \ge e_k \tag{12}$$

where $\bar{E}_k := P_A E_k$. The cut (12) is satisfied by any $z := [x^{\mathsf{T}}, \theta]^{\mathsf{T}}$ such that $Pz = [\bar{x}^{\mathsf{T}}, \bar{\theta}]^{\mathsf{T}}$ is feasible for (9) (see Lemma 1 below) and violated by $[\bar{x}_k^{\mathsf{T}}, \theta_k]^{\mathsf{T}}$.

Subcase 2.2.2. The oracle decides $\mathcal{Q}(\bar{x}_k) \le \bar{\theta}_k$.

In this case $[\bar{x}_k^{\mathsf{T}}, \bar{\theta}_k]^{\mathsf{T}}$ is feasible for (9) and the oracle return θ'_k the value of the objective function in the lp (7) at any feasible point with $\theta'_k \le \bar{\theta}_k$. We have the upper bound $Z_k := (c)^{\mathsf{T}} \bar{x}_k + \theta'_k \le (c)^{\mathsf{T}} \bar{x}_k + \bar{\theta}_k$ and we add the cut

$$\bar{c}^{\mathsf{T}} x + \theta \le Z_k \tag{13}$$

where $\bar{c} := P_A c$. Note that (13) is satisfied by all $[x^{\mathsf{T}}, \theta]^{\mathsf{T}}$ such that $Pz = [\bar{x}^{\mathsf{T}}, \bar{\theta}]^{\mathsf{T}}$ is feasible for (9) with objective function value not greater than Z_k.

In the sequel, we refer to a cut of the form (13) as the objective cut, and cuts of the form (10), (11) and (12) as nonobjective cuts. The cuts (10), (11), (12) and (13) can all be expressed in the form $a_k^{\mathsf{T}} z \le c_k$ for appropriate $a_k \in \mathbb{R}^{n_1+1}$ and $c_k \in \mathbb{R}$. We then construct the minimum volume ellipsoid $\mathcal{E}_{k+1} := \{z \in \mathbb{R}^{n_1+1} : (z - z_{k+1})^{\mathsf{T}} B_{k+1}^{-1}(z - z_{k+1}) \le 1\}$ defined by center z_{k+1} and symmetric, positive definite $B_{k+1} \in \mathbb{R}^{(n_1+1)\times(n_1+1)}$ containing the set $\{z \in \mathcal{E}_k : a_k^{\mathsf{T}} z \le c_k\}$ using the update formulae in [12]. Specifically,

$$z_{k+1} := z_k - \tau_k B_k a_k / (a_k^{\mathsf{T}} B_k a_k)^{1/2} \tag{14}$$

$$B_{k+1} := \delta_k (B_k - \sigma_k B_k a_k a_k^{\mathsf{T}} B_k / a_k^{\mathsf{T}} B_k a_k) \tag{15}$$

where $m := n_1 + 1$, $\alpha_k := (a_k^{\mathsf{T}} z_k - c_k)/(a_k^{\mathsf{T}} B_k a_k)^{1/2}$, $\tau_k := (1 + m\alpha_k)/(m+1)$, $\delta_k := (1 - \alpha_k^2)m^2/(m^2 - 1)$, and $\sigma_k := 2(1 + m\alpha_k)/[(1 + \alpha_k)(m+1)]$.

It is easy to show that the cuts (10), (11), (12) and if $Z_k < c^{\mathsf{T}} \bar{x}_k + \bar{\theta}_k$ the cut (13) are 'deep' in the sense of [12]. We also have from [12] that

$$\frac{\mathrm{Vol}(\mathcal{E}_{k+1})}{\mathrm{Vol}(\mathcal{E}_k)} < \exp\left[-\frac{1}{2(n_1+2)}\right]. \tag{16}$$

As we shall see in §5, the work of the oracle may be performed in many ways. We have thus described the following class of ellipsoid algorithms for (9).

Algorithm Class 1.

Initialization:

choose $x_0 \in \mathbb{R}^{n_1}$, $\theta_0 \in \mathbb{R}$ and $L > 0$ such that with $z_0 := [x_0^{\mathsf{T}}, \theta_0]^{\mathsf{T}}$, $B_0 := 2^{2L} I \in \mathbb{R}^{(n_1+1)\times(n_1+1)}$ and $\mathcal{E}_0 := \{z \in \mathbb{R}^{n_1+1} : (z - z_0)^{\mathsf{T}} B_0^{-1}(z - z_0) \le 1\}$, a minimizer z^* of (9) is in $\mathrm{int}(\mathcal{E}_0)$.

Main Step:

```
begin
  for $k = 0, 1, \ldots$ do
    $\bar{z}_k := [\bar{x}_k^{\mathsf{T}}, \theta_k]^{\mathsf{T}} := P z_k$;
```

```
        **if** $\bar{x}_k \notin S_2$ **then**
            choose $j_k$ such that $e_{j_k}^{\mathsf{T}}\bar{x}_k < 0$;
            $\bar{e}_{j_k} := P_A e_{j_k}$;
            $a_k := [-\bar{e}_{j_k}^{\mathsf{T}}, 0]^{\mathsf{T}}$; $c_k := 0$;
        **else**
            call the oracle;
            **if** oracle decides $\bar{x}_k \notin S_3$ **then**
                $\bar{D}_k := P_A D_k$;
                $a_k := [-\bar{D}_k^{\mathsf{T}}, 0]^{\mathsf{T}}$; $c_k := -d_k$;
            **else**
                **if** the oracle decides $\mathcal{Q}(\bar{x}_k) > \bar{\theta}_k$ **then**
                    $\bar{E}_k := P_A E_k$;
                    $a_k := [-\bar{E}_k^{\mathsf{T}}, -1]^{\mathsf{T}}$; $c_k := -e_k$;
                **else**
                    $Z_k := c^{\mathsf{T}}\bar{x}_k + \theta'_k$;
                    $a_k := [\bar{c}^{\mathsf{T}}, 1]^{\mathsf{T}}$; $c_k := Z_k$;
                **end if**;
            **end if**;
        **end if**;
        update $z_k$ and $B_k$ to $z_{k+1}$ and $B_{k+1}$ respectively using (14,15);
    **end for**;
**end**
```

2.2 The complexity of the algorithms

We now analyze the complexity of Algorithm Class 1. In order to do that we need the following definitions. Let

$$S_P := \left\{ \begin{bmatrix} x \\ \theta \end{bmatrix} \in \mathbb{R}^{n_1+1} : P \begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} P_A x \\ \theta \end{bmatrix} \text{ is feasible for (9)} \right\},$$

and given a tolerance $\epsilon > 0$ let

$$S_\epsilon := \left\{ \begin{bmatrix} x \\ \theta \end{bmatrix} \in \mathbb{R}^{n_1+1} : \begin{bmatrix} x \\ \theta \end{bmatrix} \in S_P, c^{\mathsf{T}}\bar{x} + \bar{\theta} \le c^{\mathsf{T}}x^* + \theta^* + \epsilon \right\}. \tag{17}$$

Further, if $\{z_k := [x_k^{\mathsf{T}}, \theta_k]^{\mathsf{T}}\}$ is the sequence generated by Algorithm 1, define

$$\hat{Z}_{k+1} := \begin{cases} Z_{k+1} \text{ if } Z_{k+1} < \hat{Z}_k \\ \hat{Z}_k \quad \text{otherwise,} \end{cases} \tag{18}$$

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{\theta}_{k+1} \end{bmatrix} := \begin{cases} \begin{bmatrix} \hat{x}_k \\ \hat{\theta}_k \end{bmatrix} & \text{if } \hat{Z}_{k+1} = \hat{Z}_k \\ \begin{bmatrix} \bar{x}_{k+1} \\ \bar{\theta}_{k+1} \end{bmatrix} & \text{otherwise,} \end{cases} \tag{19}$$

for $k = 0, 1, \ldots$ where $\hat{x}_0 := \bar{x}_0$, $\hat{\theta}_0 := \bar{\theta}_0$ and $Z_0 := +\infty$, $\hat{Z}_0 := +\infty$. Note that $[\hat{x}_k^{\mathsf{T}}, \hat{\theta}_k]^{\mathsf{T}}$ is the best solution to (9), given the iterates $0, 1, \ldots, k$ generated by the algorithm.

We need the following two lemmas before we present our main result on the complexity of members of Algorithm Class 1.

Lemma 1 *The cuts (10), (11) and (12) are satisfied by every element of S_P.*

PROOF. Take any $[x^{\mathsf{T}}, \theta]^{\mathsf{T}} \in S_P$. We have $\bar{e}_{j_k}^{\mathsf{T}} x = e_{j_k}^{\mathsf{T}}(P_A x) \geq 0$ for the cut (10), $\bar{D}_k^{\mathsf{T}} x = D_k^{\mathsf{T}}(P_A x) \geq d_k$ for the cut (11), and $\bar{E}_k^{\mathsf{T}} x + \theta = E_k^{\mathsf{T}}(P_A x) + \theta \geq e_k$ for the cut (12). $\square$

Lemma 2 *Suppose that $[\hat{x}_k^{\mathsf{T}}, \hat{\theta}_k]^{\mathsf{T}} \notin S_\epsilon$ for some $k \geq 0$. Then every element of S_ϵ satisfies the cut (13).*

PROOF. For any $[x^{\mathsf{T}}, \theta]^{\mathsf{T}} \in S_\epsilon$, we have

$$\begin{aligned} \bar{c}^{\mathsf{T}} x + \theta &= c^{\mathsf{T}}(P_A x) + \theta \\ &= c^{\mathsf{T}} \bar{x} + \bar{\theta} \\ &\leq c^{\mathsf{T}} x^* + \theta^* + \epsilon \leq Z_k. \end{aligned}$$

The first inequality is due to the fact that $[x^{\mathsf{T}}, \theta]^{\mathsf{T}} \in S_\epsilon$, and the second inequality is true because $[\hat{x}_k^{\mathsf{T}}, \hat{\theta}_k]^{\mathsf{T}} \notin S_\epsilon$ and consequently $[\bar{x}_k^{\mathsf{T}}, \bar{\theta}_k] \notin S_\epsilon$. $\square$

We now have our main result of this section.

Theorem 3 *Suppose that the set $S_\epsilon \cap \mathcal{E}_0$ contains a ball of radius 2^{-L}. Let $N := 4(n_1 + 1)(n_1 + 2)L \ln 2$. Let the sequence $\{[x_k^{\mathsf{T}}, \theta_k]^{\mathsf{T}}\}$ be generated by a member of Algorithm Class 1 and let the sequence $\{[\hat{x}_k^{\mathsf{T}}, \hat{\theta}_k]^{\mathsf{T}}\}$ be defined by (18,19). Then, $[\hat{x}_k^{\mathsf{T}}, \hat{\theta}_k]^{\mathsf{T}} \in S_\epsilon$ for all $k \geq \hat{N}$ for some $\hat{N} \leq N$. The complexity of the algorithm is $O(\omega n_1^2 L + n_1^4 L)$ arithmetic operations where ω is the cost of a call to the oracle in arithmetic operations.*

PROOF. Suppose by contradiction that $[\hat{x}_N^\mathsf{T}, \hat{\theta}_N]^\mathsf{T} \notin S_\epsilon$. Then by the derivation of the algorithm and Lemmas 1 and 2, we have $\mathcal{E}_N \supset S_\epsilon \cap \mathcal{E}_0$. So $\text{Vol}(\mathcal{E}_N) > \text{Vol}(S_\epsilon \cap \mathcal{E}_0)$ and by (16)

$$\exp\left[\frac{-N}{2(n_1+2)}\right] > \frac{\text{Vol}(\mathcal{E}_k)}{\text{Vol}(\mathcal{E}_0)} \geq \frac{\text{Vol}(S_\epsilon \cap \mathcal{E}_0)}{\text{Vol}(\mathcal{E}_0)} \geq 2^{-2(n_1+1)L}.$$

We get $N < 4(n_1+1)(n_1+2)L \ln 2 = N$, a contradiction.

Since $N = O(n_1^2 L)$ and the number of arithmetic operations per iteration of a member of Algorithm Class 1 is $O(\omega + n_1^2)$, the complexity is $O(\omega n_1^2 L + n_1^4 L)$ arithmetic operations. □

3 A class of cutting plane algorithms for L-shaped linear programs based on volumetric centers

The volumetric cutting plane method was first proposed by Vaidya [38] and was later simplified and strengthened by Anstreicher [2] for solving the convex feasibility problem. In the sequel we show how the volumetric cutting plane algorithm described in [2] may be modified for problem (9). We also show that in terms of the number of iterations its complexity is $O(n_1 L)$ which is a factor of n_1 less than that of the ellipsoid method.

Let $\mathcal{P}$ be the bounded full-dimensional polytope

$$\mathcal{P} := \{z \in \mathbb{R}^{(n_1+1)} : \Lambda z \geq \beta\} \tag{20}$$

where $\Lambda \in \mathbb{R}^{m \times (n_1+1)}$ and $\beta \in \mathbb{R}^m$. For simplicity, we denote $\mathcal{P}$ by $\mathcal{P} = [\Lambda, \beta]$. The volumetric center w of $\mathcal{P}$ is the point that minimizes the volumetric barrier $V : \text{int}(\mathcal{P}) \to \mathbb{R}$ defined by $V(z) := \frac{1}{2}\ln[\det(\Lambda^\mathsf{T} S(z)^{-2}\Lambda)]$ where $s(z) := \Lambda z - \beta > 0$ and $S(z) := \text{diag}(s(z))$. For z with $s = s(z) > 0$, define $P(s) := S^{-1}\Lambda(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\Lambda^\mathsf{T} S^{-1}$ and let $P^{(2)} = P \circ P$, where $\circ$ denotes the Schur product (i.e., $P_{ij}^{(2)} = P_{ij}^2$, $\forall i,j$). Define $\sigma_i := P_{ii}$ for $i = 1, 2, \ldots, m$ and $\Sigma := \text{diag}([\sigma_1, 2, \ldots, \sigma_m]^\mathsf{T})$. Then the gradient g and Hessian H of V at z are given by

$$\begin{aligned} g &:= g(z) = \nabla V(z) = -\Lambda^\mathsf{T} S^{-1}\sigma \quad \text{and} \\ H &:= H(z) = \nabla^2 V(z) = \Lambda^\mathsf{T} S^{-1}(3\Sigma - 2P^{(2)})S^{-1}\Lambda. \end{aligned}$$

Note that

$$\sigma_i(z) = \frac{\alpha_i^{\mathsf{T}}(\Lambda^{\mathsf{T}} S^{-2}\Lambda)^{-1}\alpha_i}{(\alpha_i^{\mathsf{T}} z - \beta_i)^2}, \quad i = 1, 2, \ldots, m. \tag{21}$$

Define $\mu = \mu(z) := (2\sqrt{\sigma_{min}} - \sigma_{min})^{-1/2}$ where $\sigma_{min} = \sigma_{min}(z) = \min_{1 \le i \le m} \sigma_i(z)$.

First we state some results on the volumetric barrier function that we need in the sequel. Their proofs can be found in Anstreicher [2] or Vaidya [38].

Lemma 4 *For any $z \in \mathcal{P}$, we have*

(a) $\sigma_i \le 1$, $\forall i$.

(b) $\sum_{i=1}^m \sigma_i = n_1 + 1$.

PROOF. See Vaidya [38]. $\square$

Let $Q = Q(z) := \Lambda^{\mathsf{T}} S^{-2} \Sigma \Lambda$, then $Q(z)$ is a good approximation of $H(z)$ in the sense that

$$Q(z) \preceq H(z) \preceq 3Q(z).$$

where $A \preceq B$ iff $B - A$ is positive semidefinite for symmetric positive semidefinite A, B.

Let $p = p(z) := -H^{-1}g$ denote the Newton direction for V at z. We will use the notation that $\hat{z} := z + p$, $\hat{s} := s(\hat{z})$, $\hat{g} := g(\hat{z})$, $\hat{p} := p(\hat{z})$, $\hat{H} := H(\hat{z})$, $\hat{Q} := Q(\hat{z})$. Also, for symmetric positive definite matrix A and vector z define $||z||_A := \sqrt{z^{\mathsf{T}} A z}$.

Lemma 5 *Let z have $s = s(z) > 0$. Assume that $\mu ||p||_H \le .014$, and let $\hat{z} = z + p$, then*

(a) $||\hat{p}||_{\hat{H}} \le 21.6\mu ||p||_H^2$,

(b) $\hat{\mu} ||\hat{p}||_{\hat{H}} \le 21.6\mu^2 ||p||_H^2$,

(c) V has a unique minimizer w in the interior of $\mathcal{P}$, and $V(z) - V(w) \le 1.11 ||p||_H^2$.

PROOF. See [2, Lemma 2.4] and [2, Theorem 2.6]. $\square$

3.1 The description of algorithms

Our algorithm begins with the simplex

$$\mathcal{P}^0 := [\Lambda^0, \beta^0] \text{ with } \Lambda^0 := \begin{bmatrix} I_{n_1+1} \\ -e^\mathsf{T} \end{bmatrix}, \ \beta^0 := \begin{bmatrix} -2^L e \\ -(n_1+1)2^L \end{bmatrix} \tag{22}$$

containing a minimizer $z^* := [(x^*)^\mathsf{T}, \theta^*]^\mathsf{T}$ of (9), where $e = (1,1,\dots,1)^\mathsf{T} \in \mathbb{R}^{n_1+1}$. As indicated in [2] the volumetric center z_0 of $\mathcal{P}^0$ is given by

$$z_0 := (\frac{n_1}{n_1+2})2^L e, \tag{23}$$

and the value of V at z_0 is $V^0(z_0) = -\ln 2(n_1+1)(L+1) + (n_1+1)\ln[1 + 1/(n_1+1)] + \ln(2+n_1)$.

At the beginning of iteration $k(\geq 0)$ we have a point $z_k \in \mathbb{R}^{n_1+1}$, and a polyhedral set

$$\mathcal{P}^k := [\Lambda^k, \beta^k] \tag{24}$$

where $\Lambda^k \in \mathbb{R}^{m_k \times (n_1+1)}$ and $\beta^k \in \mathbb{R}^{m_k}$. Now we compute $\sigma^k_{min} := \sigma_{min}(z_k)$ and consider the following two cases where $\bar{\sigma} \in (0,1)$ is a given constant.

Case 1. $\sigma^k_{min} \geq \bar{\sigma}$.

In this case a constraint is added or translated. Let the projection $\bar{z}_k := [\bar{x}_k^\mathsf{T}, \bar{\theta}_k]^\mathsf{T} = Pz_k$ so that $\bar{x}_k = P_A x_k \in S_1$ and $\bar{\theta}_k = \theta_k$.

Subcase 1.1. $\bar{x}_k \notin S_2$.

Then there exists j_k with $1 \leq j_k \leq n_1$ such that $e_{j_k}^\mathsf{T} \bar{x}_k < 0$. We let $\alpha_k := [\bar{e}_{j_k}^\mathsf{T}, 0]^\mathsf{T}$, where $\bar{e}_{j_k}^\mathsf{T} = P_A e_{j_k}$.

Subcase 1.2. $\bar{x}_k \in S_2$.

So, $A\bar{x}_k = 0$ and $\bar{x}_k \geq 0$ and we call the oracle.

Subcase 1.2.1. The oracle indicates that $\bar{x}_k \notin S_3$.

In this case as we indicated in §1, the oracle returns $D_k \in \mathbb{R}^{n_1}$ and $d_k \in \mathbb{R}$ such that $D_k^\mathsf{T} x \geq d_k$ for all $x \in S_3$ and $D_k^\mathsf{T} \bar{x}_k < d_k$. We let $\alpha_k := [\bar{D}_k^\mathsf{T}, 0]^\mathsf{T}$, where $\bar{D}_k = P_A D_k$.

Subcase 1.2.2. The oracle indicates that $\bar{x}_k \in S_3$.

In this case as indicated in §1, the oracle decides whether $\mathcal{Q}(\bar{x}_k) \leq \bar{\theta}_k$ or not. We have two further subcases.

Subcase 1.2.2.1. The oracle decides $\mathcal{Q}(\bar{x}_k) > \bar{\theta}_k$.

In this case $[\bar{x}_k^\mathsf{T}, \bar{\theta}_k]^\mathsf{T}$ is not feasible for (9) and the oracle returns $E_k \in \mathbb{R}^{n_1}$ and $e_k \in \mathbb{R}$ such that for all $[x^\mathsf{T}, \theta]^\mathsf{T}$ feasible for (9) $E_k^\mathsf{T} x + \theta \geq e_k$ and $E_k^\mathsf{T} \bar{x}_k + \bar{\theta}_k < e_k$. We let $\alpha_k := [\bar{E}_k^\mathsf{T}, 1]^\mathsf{T}$, where $\bar{E}_k = P_A E_k$.

Subcase 1.2.2.2. The oracle decides $\mathcal{Q}(\bar{x}_k) \leq \bar{\theta}_k$.

In this case $[\bar{x}_k^{\mathsf{T}}, \bar{\theta}_k]^{\mathsf{T}}$ is feasible for (9). We let $\alpha_k := [-\bar{c}^{\mathsf{T}}, -1]^{\mathsf{T}}$, where $\bar{c} = P_A c$.

We now describe how we update $\mathcal{P}^k = [\Lambda^k, \beta^k]$ after addition or translation of a cut.

(a) If α_k is not used in constructing Λ^k, then we add the cut

$$\alpha_k^{\mathsf{T}} z \geq \beta_{m_k+1}^k \tag{25}$$

with $\beta_{m_k+1}^k$ satisfying $\alpha_k^{\mathsf{T}} z_k > \beta_{m_k+1}^k$ and

$$\frac{\alpha_k^{\mathsf{T}}((\Lambda^k)^{\mathsf{T}}(S^k)^{-2}\Lambda^k)^{-1}\alpha_k}{(\alpha_k^{\mathsf{T}} z_k - \beta_{m_k+1}^k)^2} = \tau \tag{26}$$

where $\tau > 0$ is a suitably chosen constant. Let $m_{k+1} := m_k + 1$,

$$\Lambda^{k+1} := \begin{bmatrix} \Lambda^k \\ (\alpha^k)^{\mathsf{T}} \end{bmatrix}, \text{ and } \beta^{k+1} := \begin{bmatrix} \beta^k \\ \beta_{m_k+1}^k \end{bmatrix}.$$

(b) If α_k is already used in constructing Λ, let $\alpha_{j_k}^{\mathsf{T}} z \geq \beta_{j_k}^k$ be the corresponding constraint in the system. Then we translate it to the cut

$$\alpha_{j_k}^{\mathsf{T}} z \geq \tilde{\beta}_{j_k}^k \tag{27}$$

with $\tilde{\beta}_{j_k}^k$ satisfying $\alpha_{j_k}^{\mathsf{T}} z_k > \tilde{\beta}_{j_k}^k > \beta_{j_k}^k$ and

$$\alpha_{j_k}^{\mathsf{T}}((\Lambda^k)^{\mathsf{T}}(S^k)^{-2}\Lambda^k)^{-1}\alpha_{j_k}\left(\frac{1}{(\alpha_{j_k}^{\mathsf{T}} z_k - \tilde{\beta}_{j_k}^k)^2} - \frac{1}{(\alpha_{j_k}^{\mathsf{T}} z_k - \beta_{j_k}^k)^2}\right) = \tau_1,$$

where $\tau_1 > 0$ is also a suitably chosen constant. Let $m_{k+1} := m_k$, and Λ^{k+1} and β^{k+1} be Λ_k and β^k respectively with the j_k-th row translated. Note that translation is especially important relative to the objective cut.

Case 2. $\sigma_{min}^k < \bar{\sigma}$.

In this case a constraint is removed. Let $\sigma_{j_k}^k := \sigma_{min}^k$. Then the constraint

$$(e_{j_k}^{\mathsf{T}}\Lambda^k)z \geq \beta_{j_k}^k \tag{28}$$

is deleted. Let $m_{k+1} := m_k - 1$, and Λ^{k+1} and β^{k+1} be Λ_k and β^k respectively with the j_k-th row removed.

When a constraint is added, translated or deleted the volumetric center shifts. A sequence of pure Newton steps is performed beginning with the current iterate z_k to obtain the next iterate z_{k+1} ‘sufficiently close’ to the volumetric center w_{k+1} of the new polyhedral set. The 'closeness' of z_{k+1} to w_{k+1} is measured in terms of the closeness of $V^{k+1}(z_{k+1})$ and $V^{k+1}(w_{k+1})$. Specifically, we use $V^{k+1}(z_{k+1}) - V^{k+1}(w_{k+1}) \leq \delta$ where $\delta \in (0, 1)$ is a given constant.

Our discussion above leads to the following class of volumetric center cutting plane algorithms for (9).

Algorithm Class 2.

Initialization:

```
    choose $L > 0$ so that $z^* \in \text{int } \mathcal{P}^0$ where $\mathcal{P}^0$ is as in (22), and
    $\bar{\sigma} \in (0, 1)$, $\tau > 0$ and $\tau_1 > 0$;
    define $z_0$ by (23).
```

Main Step:

```
begin

    for $k = 0, 1, \ldots$ do
        $\sigma^k_{min} := \sigma_{min}(z_k)$;
        if $\sigma^k_{min} \geq \bar{\sigma}$ then
            $\bar{z}_k := [\bar{x}_k^\mathsf{T}, \theta_k]^\mathsf{T} := P z_k$;
            if $\bar{x}_k \notin S_2$ then
                choose $j_k$ such that $e_{j_k}^\mathsf{T} \bar{x}_k < 0$;
                $\bar{e}_{j_k} := P_A e_{j_k}$;
                $\alpha_k := [\bar{e}_{j_k}^\mathsf{T}, 0]^\mathsf{T}$;
            else
                call the oracle;
                if oracle decides $\bar{x}_k \notin S_3$ then
                    $\bar{D}_k := P_A D_k$;
                    $\alpha_k := [\bar{D}_k^\mathsf{T}, 0]^\mathsf{T}$;
                else
                    if the oracle decides $\mathcal{Q}\bar{x}_k > \bar{\theta}_k$ then
                        $\bar{E}_k := P_A E_k$;
                        $\alpha_k := [\bar{E}_k^\mathsf{T}, 1]^\mathsf{T}$;
                    else
                        $\alpha_k := [-\bar{c}^\mathsf{T}, -1]^\mathsf{T}$;
                    end if;
                end if;
            end if;
            define $\Lambda^{k+1}$ and $\beta^{k+1}$ as in Case 1 above to
            add the cut (25) and let $m_{k+1} := m_k + 1$
            or translate the $j_k$-th cut to (27) and let $m_{k+1} := m_k$;
        else
```

```
            $m_{k+1} := m_k - 1$;
            define $\Lambda^{k+1}$ and $\beta^{k+1}$ as in Case 2 above to
            delete the cut (28);
        end if;
        beginning with $z_k$, take a sequence of pure Newton steps to obtain
        $z_{k+1}$ 'close' to new volumetric center;
    end for;
end
```

3.2 The complexity of the algorithms

We begin by analyzing the effect of adding, translating and deleting a cut on the volumetric barrier function. For notational simplicity we omit the subscript k and adopt the following convention. The current constraint system is given by $[\Lambda, \beta]$, where Λ is a $m \times (n_1 + 1)$ matrix. Let z be the current point with $s = s(z) > 0$. We use $[\tilde{\Lambda}, \tilde{\beta}]$ to express the constraint system after the addition, translation or deletion of a cut.

(i) Adding a constraint

Lemma 6 *Suppose that a constraint $(\alpha_{m+1}^{\mathsf{T}}, \beta_{m+1})$ is added in the Algorithm. Then*

(a) $\tilde{V}(z) = V(z) + (1/2)\ln(1+\tau)$,
(b) $\tilde{\sigma}_{m+1} = \tau/(1+\tau)$ *and* $\sigma_i \geq \tilde{\sigma}_i \geq \sigma_i/(1+\tau)$, $i = 1, 2, \ldots, m$, *and*
(c) $||\tilde{p}||_{\tilde{H}} \leq \sqrt{1+\tau}[\sqrt{3}||p||_H + \tau(1 + \sqrt{\tau/\bar{\sigma}})/(1+\tau)]$.

PROOF. See [2, Lemma 4.1], [2, Lemma 4.2] and [2, Theorem 4.3]. □

Lemma 7 *Suppose that the point z has $s = s(z) > 0$, and $||p||_H \leq r_1 =$ .0007, $\mu||p||_H \leq r_2 =$ .0035, and that $\bar{\sigma} =$ .0004. Suppose that a constraint $(\alpha_{m+1}^{\mathsf{T}}, \beta_{m+1})$ is added with $\tau =$ .00056. In the new constraint system $[\tilde{\Lambda}, \tilde{\beta}]$, after one Newton step, i.e., $\hat{z} = z + \tilde{p}, \tilde{p} = \tilde{p}(z) = \tilde{H}^{-1}\tilde{g}$, we have*

(a) $||\hat{p}||_{\hat{H}} \leq r_1$,
(b) $\hat{\mu}||\hat{p}||_{\hat{H}} \leq r_2$, *and*
(c) $\tilde{V}(\hat{z}) \geq V(z) + \triangle V^+$, *where* $\triangle V^+ =$ .000273.

PROOF. See [2, Theorem 6.1]. □

Lemma 7 implies that the Algorithm requires only a single Newton step after a constraint addition.

(ii) Translating a constraint
As mentioned earlier translating a cut is especially relevant for the objective cut. To the best of our knowledge the effect of translating a cut on the volmetric center has not been previously analyzed. If translation is not allowed then multiple objective cuts with *the same left-hand-side and different right-hand-sides* may be present in the constraint system. Of course that would imply maintaining redundant cuts. Since we allow translation of existing cuts, the constraint systems of volumetric center algorithms presented in this paper contain at most one objective cut at any given time.

Without loss of generality we assume that the m-th constraint is translated. Let $\tilde{s}_m := \alpha_m^\mathsf{T} z - \tilde{\beta}_m$. Then

$$\tau_1 = (\frac{1}{\tilde{s}_m^2} - \frac{1}{s_m^2})\alpha_m^\mathsf{T}(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_m. \tag{29}$$

We note that $\tilde{s}_m < s_m$ and $\tilde{\beta}_m > \beta_m$.

Lemma 8 *Suppose that a constraint* $(\alpha_m^\mathsf{T}, \beta_m)$ *is translated to* $(\alpha_m^\mathsf{T}, \tilde{\beta}_m)$*, and* $\tau_1 > 0$ *is given by (29). Then*

(a) $\tilde{V}(z) = V(z) + (1/2)\ln(1+\tau_1)$,
(b) $\tilde{\sigma}_m = \tau_1/(1+\tau_1) + \sigma_m/(1+\tau_1) \geq \sigma_m/(1+\tau_1)$ *and* $\sigma_i \geq \tilde{\sigma}_i \geq \sigma_i/(1+\tau_1)$, $i = 1, 2, \ldots, m-1$, *and*
(c) $||\tilde{p}||_{\tilde{H}} \leq \sqrt{1+\tau_1}[\sqrt{3}||p||_H + \tau_1/(1+\tau_1) + \tau_1/(2\sqrt{\bar{\sigma}}) + \tau_1(1-\bar{\sigma})/\sqrt{\bar{\sigma}(1+\tau_1)}]$.

PROOF. (a) Note that

$$\tilde{\Lambda}^\mathsf{T}\tilde{S}^{-2}\tilde{\Lambda} = \Lambda^\mathsf{T} S^{-2}\Lambda + (\frac{1}{\tilde{s}_m^2} - \frac{1}{s_m^2})\alpha_m\alpha_m^\mathsf{T}. \tag{30}$$

Therefore,

$$\begin{aligned}\det(\tilde{\Lambda}^\mathsf{T}\tilde{S}^{-2}\tilde{\Lambda}) &= \det(\Lambda^\mathsf{T} S^{-2}\Lambda)\cdot\det(I + (\frac{1}{\tilde{s}_m^2} - \frac{1}{s_m^2})(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_m\alpha_m^\mathsf{T}) \\ &= \det(\Lambda^\mathsf{T} S^{-2}\Lambda)\cdot(1 + (\frac{1}{\tilde{s}_m^2} - \frac{1}{s_m^2})\alpha_m^\mathsf{T}(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_m) \\ &= \det(\Lambda^\mathsf{T} S^{-2}\Lambda)(1+\tau_1)\end{aligned}$$

which yields

$$\tilde{V}(z) = \frac{1}{2}\ln[\det(\tilde{\Lambda}^\mathsf{T}\tilde{S}^{-2}\tilde{\Lambda})]$$

$$
\begin{aligned}
&= \frac{1}{2}\ln[\det(\Lambda^\mathsf{T} S^{-2}\Lambda)] + \frac{1}{2}\ln(1+\tau_1)\\
&= V(z) + \frac{1}{2}\ln(1+\tau_1).
\end{aligned}
$$

(b) From (30), we obtain

$$
\begin{aligned}
(\tilde{\Lambda}^\mathsf{T}\tilde{S}^{-2}\tilde{\Lambda})^{-1} = &(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1} -\\
&(\frac{1}{\tilde{s}_m^2} - \frac{1}{s_m^2})\frac{(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_m\alpha_m^\mathsf{T}(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}}{1+\tau_1}.
\end{aligned}
\tag{31}
$$

(31) with $\tilde{\sigma}_i = (1/s_i^2)\alpha_i^\mathsf{T}(\tilde{\Lambda}^\mathsf{T}\tilde{S}^{-2}\tilde{\Lambda})^{-1}\alpha_i$ for $i = 1, 2, \ldots, m-1$ yields

$$
\tilde{\sigma}_i = \sigma_i - \frac{1}{s_i^2}(\frac{1}{\tilde{s}_m^2} - \frac{1}{s_m^2})\frac{(\alpha_i^\mathsf{T}(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_m)^2}{1+\tau_1},
$$

which implies that $\sigma_i \geq \tilde{\sigma}_i$ for $i = 1, 2, \ldots, m-1$. Then it follows from

$$
\begin{aligned}
&\frac{1}{s_i^2}(\frac{1}{\tilde{s}_m^2} - \frac{1}{s_m^2})(\alpha_i^\mathsf{T}(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_m)^2\\
\leq &\frac{1}{s_i^2}(\alpha_i^\mathsf{T}(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_i)(\frac{1}{\tilde{s}_m^2} - \frac{1}{s_m^2})(\alpha_m^\mathsf{T}(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_m)\\
= &\sigma_i\tau_1
\end{aligned}
$$

that $\tilde{\sigma}_i \geq \sigma_i - \sigma_i\tau_1/(1+\tau_1) = \sigma_i/(1+\tau_1)$ for $i = 1, 2, \ldots, m-1$. For $i = m$, from (31) we have

$$
(\frac{1}{\tilde{s}_m^2} - \frac{1}{s_m^2})\alpha_m^\mathsf{T}(\tilde{\Lambda}^\mathsf{T}\tilde{S}^{-2}\tilde{\Lambda})^{-1}\alpha_m = \tau_1 - \frac{\tau_1^2}{1+\tau_1} = \frac{\tau_1}{1+\tau_1}
$$

and

$$
\frac{1}{s_m^2}\alpha_m^\mathsf{T}(\tilde{\Lambda}^\mathsf{T}\tilde{S}^{-2}\tilde{\Lambda})^{-1}\alpha_m = \sigma_m - \frac{\tau_1\sigma_m}{1+\tau_1} = \frac{\sigma_m}{1+\tau_1}.
$$

Adding the above two inequalities and using $\tilde{\sigma}_m = (1/\tilde{s}_m^2)\alpha_m^\mathsf{T}(\tilde{\Lambda}^\mathsf{T}\tilde{S}^{-2}\tilde{\Lambda})^{-1}\alpha_m$ we get that

$$
\tilde{\sigma}_m = \frac{\sigma_m}{1+\tau_1} + \frac{\tau_1}{1+\tau_1}. \tag{32}
$$

(c) As in the proof of [2] and using $\tilde{s}_i = s_i$ for $i = 1, 2, \ldots, m-1$, and $\tilde{s}_m < s_m$, we have

$$\tilde{Q} = \sum_{i=1}^{m} \frac{\tilde{\sigma}_i}{\tilde{s}_i^2} \alpha_i \alpha_i^{\mathsf{T}} \succeq \sum_{i=1}^{m} \frac{\tilde{\sigma}_i}{s_i^2} \alpha_i \alpha_i^{\mathsf{T}} \succeq \frac{1}{1+\tau_1} \sum_{i=1}^{m} \frac{\sigma_i}{s_i^2} \alpha_i \alpha_i^{\mathsf{T}} = \frac{1}{1+\tau_1} Q.$$

So $\tilde{Q}^{-1} \preceq (1+\tau_1) Q^{-1}$, and then

$$||\tilde{p}||_{\tilde{H}} = ||\tilde{g}||_{\tilde{H}^{-1}} \leq ||\tilde{g}||_{\tilde{Q}^{-1}} \leq \sqrt{1+\tau_1} ||\tilde{g}||_{Q^{-1}} = \sqrt{1+\tau_1} ||\tilde{\Lambda}^{\mathsf{T}} \tilde{S}^{-1} \tilde{\sigma}||_{Q^{-1}}.$$

Since

$$\begin{aligned} \tilde{\Lambda}^{\mathsf{T}} \tilde{S}^{-1} \tilde{\sigma} &= \sum_{i=1}^{m} \frac{\tilde{\sigma}_i}{\tilde{s}_i} \alpha_i \\ &= \sum_{i=1}^{m} \frac{\sigma_i}{s_i} \alpha_i + \sum_{i=1}^{m-1} \frac{\tilde{\sigma}_i - \sigma_i}{s_i} \alpha_i + \frac{1}{\tilde{s}_m} (\tilde{\sigma}_m - \sigma_m) \alpha_m + \sigma_m (\frac{1}{\tilde{s}_m} - \frac{1}{s_m}) \alpha_m, \end{aligned}$$

we get with $g = -\Lambda^{\mathsf{T}} S^{-1} \sigma$ that

$$\begin{aligned} ||\tilde{p}||_{\tilde{H}} \leq& \sqrt{1+\tau_1} (||g||_{Q^{-1}} + || \sum_{i=1}^{m-1} \frac{\tilde{\sigma}_i - \sigma_i}{s_i} \alpha_i ||_{Q^{-1}} \\ &+ || \frac{1}{\tilde{s}_m} (\tilde{\sigma}_m - \sigma_m) \alpha_m ||_{Q^{-1}} + || \sigma_m (\frac{1}{\tilde{s}_m} - \frac{1}{s_m}) \alpha_m ||_{Q^{-1}}) \end{aligned}$$

As in the proof of [2, Theorem 4.3], we have

$$|| \sum_{i=1}^{m-1} \frac{\tilde{\sigma}_i - \sigma_i}{s_i} \alpha_i ||_{Q^{-1}} \leq \frac{\tau_1}{1+\tau_1} \tag{33}$$

and

$$||g||_{Q^{-1}} \leq \sqrt{3} ||p||_H. \tag{34}$$

The fact that $\sigma_{min} \geq \bar{\sigma}$ implies that $Q^{-1} = (\Lambda^{\mathsf{T}} S^{-2} \Sigma \Lambda)^{-1} \preceq (1/\bar{\sigma})(\Lambda^{\mathsf{T}} S^{-2} \Lambda)^{-1}$. Therefore,

$$|| \frac{1}{\tilde{s}_m} (\tilde{\sigma}_m - \sigma_m) \alpha_m ||_{Q^{-1}}^2 \leq \bar{\sigma}^{-1} (\tilde{\sigma}_m - \sigma_m)^2 \frac{1}{\tilde{s}_m^2} \alpha_m^{\mathsf{T}} (\Lambda^{\mathsf{T}} S^{-2} \Lambda)^{-1} \alpha_m.$$

Using $\tilde{\sigma}_m - \sigma_m = [\tau_1/(1+\tau_1)](1-\sigma_m)$ which follows from (32), and $(1/\tilde{s}_m^2)\alpha_m^\mathsf{T}(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_m = (1+\tau_1)\tilde{\sigma}_m \le 1+\tau_1$ which follows from (31), we get

$$||\frac{1}{\tilde{s}_m}(\tilde{\sigma}_m - \sigma_m)\alpha_m||_{Q^{-1}}^2 \le \frac{\tau_1^2}{\bar{\sigma}(1+\tau_1)}(1-\sigma_m)^2 \le \frac{\tau_1^2}{\bar{\sigma}(1+\tau_1)}(1-\bar{\sigma})^2. \quad (35)$$

Also

$$\begin{aligned} ||\sigma_m(\frac{1}{\tilde{s}_m} - \frac{1}{s_m})\alpha_m||_{Q^{-1}}^2 \le & \bar{\sigma}^{-1}\sigma_m^2(\frac{1}{\tilde{s}_m} - \frac{1}{s_m})^2\alpha_m^\mathsf{T}(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_m \\ = & \bar{\sigma}^{-1}\sigma_m\frac{1}{s_m^2}(\frac{1}{\tilde{s}_m} - \frac{1}{s_m})^2(\alpha_m^\mathsf{T}(\Lambda^\mathsf{T} S^{-2}\Lambda)^{-1}\alpha_m)^2. \end{aligned}$$

Now using $1/s_m^2 \le 1/(s_m\tilde{s}_m) \le (1/4)(1/\tilde{s}_m + 1/s_m)^2$, we get

$$||\sigma_m(\frac{1}{\tilde{s}_m^2} - \frac{1}{s_m^2})\alpha_m||_{Q^{-1}}^2 \le \frac{\tau_1^2}{4\bar{\sigma}}. \quad (36)$$

Combining (33), (34), (35) and (36), we have the result in part (c). □

Lemma 9 *Suppose that the point z has $s = s(z) > 0$, and $||p||_H \le r_1, \mu||p||_H \le r_2$, where r_1, r_2, and $\bar{\sigma}$ are as in Lemma 7. Suppose that a constraint $(\alpha_m^\mathsf{T}, \beta_m)$ is translated to $(\alpha_m^\mathsf{T}, \tilde{\beta}_m)$ with $\tau_1 = .000017$. In the new constraint system $[\tilde{\Lambda}, \tilde{\beta}]$, after one Newton step, i.e., $\hat{z} = z + \tilde{p}, \tilde{p} = \tilde{p}(z) = \tilde{H}^{-1}\tilde{g}$, we have*

(a) $||\hat{p}||_{\hat{H}} \le r_1$,
(b) $\hat{\mu}||\hat{p}||_{\hat{H}} \le r_2$, *and*
(c) $\tilde{V}(\hat{z}) \ge V(z) + \triangle V'$, *where* $\triangle V' = .00000219$.

PROOF. Since $\sigma_{min} \ge \bar{\sigma}$, Lemma 8 imples that $\tilde{\sigma}_{min} \ge \bar{\sigma}/(1+\tau_1) \ge 0.000399$, and $\tilde{\mu} = (2\sqrt{\tilde{\sigma}_{min}} - \tilde{\sigma}_{min})^{-1/2} < 5.03$. Therefore, Lemma 8 (c) gives $||\tilde{p}||_{\tilde{H}} \le 0.00251$, which gives $\tilde{\mu}||\tilde{p}||_{\tilde{H}} \le 0.0127$. Then using Lemma 5 we obtain

$$\begin{aligned} ||\hat{p}||_{\hat{H}} \le & 21.6 \cdot 5.03(.00251)^2 < .0007 \\ \hat{\mu}||\hat{p}||_{\hat{H}} \le & 21.6(5.03)^2(.00251)^2 < .0035. \end{aligned}$$

To prove part (c) we use the subgradient inequality on V and we get

$$\begin{aligned} \tilde{V}(\hat{z}) \ge & \tilde{V}(z) + \tilde{g}^\mathsf{T}\tilde{p} \\ = & V(z) + \frac{1}{2}\ln(1+\tau_1) - ||\tilde{p}||_{\tilde{H}}^2 \end{aligned}$$

$$\begin{aligned}&\geq V(z) + \frac{1}{2}\ln(1.000017) - (0.00251)^2 \\ &\geq V(z) + 0.00000219. \qquad \square\end{aligned}$$

Lemma 9 implies that the algorithm requires only one Newton step after a constraint translation.

(iii) Deleting a constraint

Lemma 10 *Suppose that a constraint $(\alpha_m^{\mathsf{T}}, \beta_m)$ is deleted, where $\sigma_m \leq \bar{\sigma}$. Then*

(a) $\tilde{V}(z) \geq V(z) + (1/2)\ln(1-\bar{\sigma})$,
(b) $\sigma_i \leq \tilde{\sigma}_i \leq \sigma_i/(1-\bar{\sigma})$, $i = 1, 2, \ldots, m-1$, *and*
(c) $||\tilde{p}||_{\tilde{H}} \leq (1/\sqrt{1-\sigma_{min}})[\sqrt{3}||p||_H + 2\sigma_{min}/(1-\sigma_{min})]$.

PROOF. See [2, Lemma 5.1], [2, Lemma 5.2] and [2, Theorem 5.3]. □

Lemma 11 *Suppose that the point z has $s = s(z) > 0$, and that $||p||_H \leq r_1, \mu||p||_H \leq r_2$. Suppose that the constraint $(\alpha_m^{\mathsf{T}}, \beta_m)$ is deleted, and let $[\tilde{\Lambda}, \tilde{\beta}]$ be the reduced constraint system. Then after one Newton step, i.e., $\hat{z} = z + \tilde{p}, \tilde{p} = \tilde{p}(z) = \tilde{H}^{-1}\tilde{g}$, we have*

(a) $||\hat{p}||_{\hat{H}} \leq r_1$,
(b) $\hat{\mu}||\hat{p}||_{\hat{H}} \leq r_2$, *and*
(c) $\tilde{V}(\hat{z}) \geq V(z) - \triangle V^-$, *where* $\triangle V^- = .000205$.

PROOF. See [2, Theorem 6.2]. □

Lemma 11 implies that the algorithm requires only one Newton step after a constraint deletion.

Let m_k be the number of the constraint in the system at iteration k. Then the boundedness of $\mathcal{P}^k$ implies that $m_k \geq n_1 + 1$. Assume that k_1, k_2 and k_3 are the number of iterations of translation, addition and deletion, respectively. Then $k_1 + k_2 + k_3 = k$ which with $m_0 = n_1 + 1$ implies that $k_2 \geq k_3$. Therefore $k_1/2 + k_2 \geq k/2$. So for all k,

$$\begin{aligned}V^k(z_k) &\geq V^0(z_0) + k_1\triangle V' + k_2\triangle V^+ - k_3\triangle V^- \\ &\geq V^0(z_0) + k_1\triangle V' + k_2\triangle V \\ &\geq V^0(z_0) + (k_1/2 + k_2)2\triangle V' \\ &\geq V^0(z_0) + k\triangle V' \qquad (37)\end{aligned}$$

where $\triangle V = \triangle V^{+} - \triangle V^{-} > 0$.

Also note that when a constraint is added, the condition $\sigma^k_{min} \geq \bar{\sigma}$ is always satisfied. So $m_k \bar{\sigma} \leq m_k \sigma^k_{min} \leq \sum_{i=1}^{m_k} \sigma^k_i = n_1 + 1$ and $m_k \leq (n_1+1)/\bar{\sigma} + 1$ for all k. Thus the number of constraints in the system is always bounded by $m_\sigma := (n_1+1)/\bar{\sigma} + 1$.

According to Lemma 1 and Lemma 2 in §2, any nonobjective cuts are satisfied by every element of S_P. Let us represent the right-hand-side of the objective cut by β^{t_k}. Then the objective cut has the form $\alpha^{\mathsf{T}} z \geq \beta^{t_k}$ where $\alpha := [-\bar{c}^{\mathsf{T}}, -1]^{\mathsf{T}}$. It is satisfied by every element of the set

$$S_{\beta^{t_k}} := \left\{ \begin{bmatrix} x \\ \theta \end{bmatrix} \in \mathbb{R}^{n_1+1} : \begin{bmatrix} x \\ \theta \end{bmatrix} \in S_P, -c^{\mathsf{T}} z - \theta \geq \beta^{t_k} \right\}.$$

Note that in the algorithm the objective constraint may also be dropped after its addition or translation. However, we have

Lemma 12 *Suppose that the projected feasible set $S_P \cap \mathcal{P}^0$ contains a ball of radius $r = O(1)$. Let $N_1 := \frac{1}{\triangle V'}((n_1+1)\ln(m_\sigma/r) + r_0 - V^0(z_0)) = O(n_1)$. Then the objective constraint will not be dropped for all $k \geq N_1$.*

PROOF. Suppose by contradiction that the objective constraint is dropped for some $k \geq N_1$. Then we have $S_P \cap \mathcal{P}^0 \subset \mathcal{P}^k$ since there is no objective constraint in the system. Thus $\mathrm{Vol}(S_P \cap \mathcal{P}^0) < \mathrm{Vol}(\mathcal{P}^k)$ which implies that

$$\begin{aligned} 1 &< \frac{\mathrm{Vol}(\mathcal{P}^k)}{\mathrm{Vol}(S_P \cap \mathcal{P}^0)} \\ &< (\frac{m_k}{r})^{n_1+1} (\det(\nabla^2 \phi^k(w_k)))^{-1/2} \\ &\leq (\frac{m_k}{r})^{n_1+1} \exp(-V^k(w_k)) \\ &\leq (\frac{m_k}{r})^{n_1+1} \exp(r_0 - V^k(z_k)) \\ &\leq (\frac{m_k}{r})^{n_1+1} \exp(r_0 - V^0(z_0) - k \nabla V'). \end{aligned}$$

The fourth inequality is from Lemma 5 (c) which gives that $V^k(z_k) - V^k(w_k) \leq 1.11 ||p||^2_H \leq 1.11 \cdot \ 0.0007^2 := r_0$, while the fifth inequality is from (37). Therefore we get

$$k < \frac{1}{\triangle V'}((n_1+1)\ln(m_\sigma/r) + r_0 - V^0(z_0)) = N_1$$

a contraction. $\square$

So when there is no more dropping of objective constraint, the objective upper bound $\{-\beta^{t_k}\}$ is a nonincreasing sequence.

Theorem 13 *Suppose that the volume of the set $S_\epsilon \cap \mathcal{P}^0$ contains a ball of radius 2^{-L}. Let $V^k_{max} := 0.7(n_1+1)L + (n_1+1)\ln(m_k) = O(n_1 L)$. Then the upper bound $-\beta^{t_k}$ satisfies that $-\beta^{t_k} \leq c^\mathsf{T} x^* + \theta^* + \epsilon$ whenever $V^k(z_k) \geq V^k_{max}$.*

PROOF. Notice that if $-\beta^{t_k} > c^\mathsf{T} x^* + \theta^* + \epsilon$, then by the derivation of the algorithm, $S_\epsilon \cap \mathcal{P}^0 \subset \mathcal{P}^k$. Therefore, $\mathrm{Vol}(S_\epsilon \cap \mathcal{P}^0) < \mathrm{Vol}(\mathcal{P}^k)$ and $2^{-(n_1+1)L} < m_k^{n_1+1}\exp(-V^k(w_k))$. Then $V^k(w_k) \leq (n_1+1)L\ln 2 + (n_1+1)\ln m_k$ and

$$\begin{aligned} V^k(z_k) &< V^k(w_k) + r_0 \\ &< (n_1+1)L\ln 2 + (n_1+1)\ln m_k + r_0 \\ &< 0.7(n_1+1)L + (n_1+1)\ln m_k = V^k_{max} \end{aligned}$$

which contradicts the hypothesis $V^k(z_k) \geq V^k_{max}$. $\square$

Note that by Theorem 13 the condition $V^k(z_k) \geq V^k_{max}$ can be used as a stopping criterion for Algorithm Class 2. If this stopping criterion is satisfied at iterate k, we have that $c^\mathsf{T}\hat{x}_k + \hat{\theta}_k \leq -\beta^{t_k} \leq c^\mathsf{T} x^* + \theta^* + \epsilon$, where $[\hat{x}_k^\mathsf{T}, \hat{\theta}_k]^\mathsf{T}$ is defined by (18, 19).

Theorem 14 *With the stopping criterion given above, Algorithm 2 terminates in at most $N_{vol} := (1/\nabla V')(V^k_{max} - V^0(z_0)) = O(n_1 L)$ iterations.*

PROOF. From (37) it follows that $V^k(z_k) \geq V^0(z_0) + k\nabla V'$. So $V^0(z_0) + k\nabla V' \geq V^k_{max}$, i.e., $k \geq (1/\nabla V')(V^k_{max} - V^0(z_0)) = N_{vol}$ implies that $V^k(z_k) \geq V^k_{max}$. $\square$

The number of arithmetic operations per iteration is $O(T + n_1^3)$, and therefore the total number of arithmetic operations is $O(Tn_1L + n_1^4 L)$. Compared to the complexity of Algorithm 1, Algorithm 2 has a fewer number of arithmetic operations, especially fewer calls to the oracle. Also, using fast matrix multiplication [13] the complexity of Algorithm 2 can be reduced to $O(Tn_1L + n_1 M(n_1)L)$ where $M(n_1) = n_1^{2.38}$. A similar reduction is not possible for Algorithm 1.

4 A class of cutting plane algorithms for L-shaped linear programs based on analytic centers

Algorithm Classes 1 and 2 in §§2 and 3 respectively are based on the following general scheme. At each iteration of the algorithm we have a member S of a class of sets with nice properties containing a minimizer of the problem. We then choose a test point, again with nice properties, and based on that test point we modify the set S to obtain another member $\bar{S}$ of the class of sets so that $\bar{S}$ contains the solution of the problem. We then repeat the procedure. The choice of the class of sets, the test point, and the modification of S to $\bar{S}$ are all important for obtaining convergent algorithms, and analyzing their complexities. In deriving Algorithm Class 1, the class of sets used is ellipsoids and the test point used is the center of the ellipsoid. Given the ellipsoid S containing a minimizer and information at the center of S, Algorithm Class 1 sets $\bar{S}$ to be the minimum volume ellipsoid containing S cut off by a suitable hyperplane. In deriving Algorithm Class 2, the class of polyhedral sets defined by a finite number of linear inequalities is used. The test point used is the volumetric center of such a polyhedral set. The modification of the polyhedral set S to obtain the new polyhedral set $\bar{S}$ involves the addition, deletion or translation of linear inequalities.

The notion of analytic center [35] of a polyhedral set is used extensively in the context of interior point algorithms. Atkinson and Vaidya [4] derived a cutting plane algorithm for the convex feasibility problem using the class of polyhedral sets and analytic centers of such sets as test points. Atkinson and Vaidya [4] also provided a complexity analysis of their algorithm. One of the nice features of this complexity analysis is that *no assumption* is made on the finiteness of the number of linear inequalities defining the polyhedral set maintained by the algorithm; in fact, this is proved as part of the analysis. Similar algorithms and their complexity analyses presented prior to the work of Atkinson and Vaidya [4] make that assumption.

In this section we show how the analytic center algorithm described in [4] may be modified for problem (9). Let $\mathcal{P}$ be the bounded full-dimensional polytope defined as in (20). The analytic center w of $\mathcal{P}$ is the unique point that minimizes the logarithmic barrier $F : \operatorname{int}(\mathcal{P}) \rightarrow \mathbb{R}$ defined by $F(z) := -\sum_{i=1}^{m} \ln(\alpha_i^\mathsf{T} z - \beta_i)$ where $\alpha_i := \Lambda^\mathsf{T} e_i$, $i = 1, 2, \ldots, m$. The gradient and Hessian of F at z are given by

$$\begin{aligned} \nabla F(z) &= -\Lambda^\mathsf{T} S(z)^{-1} e \quad \text{and} \\ \nabla^2 F(z) &= \Lambda^\mathsf{T} S(z)^{-2} \Lambda \end{aligned}$$

where $s(z) := \Lambda z - \beta > 0$ and $S(z) := \operatorname{diag}(s(z))$. For z with $s = s(z) > 0$, let

$\sigma(z)$ be defined as in (21). Define

$$\mu_i(z) := \begin{cases} 1 & 1 \le i \le 2(n_1+1) \\ (\alpha_i^\intercal z - \beta_i)/K(\alpha_i, \beta_i) & 2(n_1+1)+1 \le i \le m \end{cases}$$

where $K(\alpha_i, \beta_i)$ is as given in Algorithm Class 3 below for $i = 1, 2, \dots, m$.

4.1 The description of algorithms

Our algorithms begins with the box

$$\begin{aligned} \mathcal{P}^0 :=& \{z \in \mathbb{R}^{n_1+1} : -2^L \le z_i \le 2^L, 1 \le i \le n_1+1\} \\ =& \{z \in \mathbb{R}^{n_1+1} : \Lambda^0 z \ge \beta^0\} \end{aligned} \tag{38}$$

where

$$\Lambda^0 := \begin{bmatrix} I_{n_1+1} \\ -I_{n_1+1} \end{bmatrix}, \quad \beta^0 := -2^L \begin{bmatrix} e \\ e \end{bmatrix} \tag{39}$$

containing a minimizer $z^* := [(x^*)^\intercal, \theta^*]^\intercal$ of (9). The analytic center of $\mathcal{P}^0$ is $z_0 := 0$. Set $K(\alpha_i, \beta_i) := 2^L, 1 \le i \le 2(n_1+1)$.

At the beginning of iteration $k (\ge 0)$ we have a point $z_k \in \mathbb{R}^{n_1+1}$, and a polyhedral set $\mathcal{P}^k$ as in (24). We compute $\mu^k_{max} := \max_{1 \le i \le m_k} \mu_i(z_k)$ and consider the following two cases with the constants $\mu := 2$, $\bar{\sigma} := 0.04$ and $\tau := 1/16$.

Case 1. $\mu^k_{max} \le \mu$.

In this case a constraint is added.

We then consider the subcases as in §3 with α_k defined as in each subcase. We update $\mathcal{P}^k = [\Lambda^k, \beta^k]$ by adding a cut $\alpha_k^\intercal z \ge \beta^k_{m_k+1}$ with $\beta^k_{m_k+1}$ defined in (26), and let $m_{k+1} := m_k + 1$, $\Lambda^{k+1} := \begin{bmatrix} \Lambda^k \\ \alpha_k^\intercal \end{bmatrix}$, $\beta^{k+1} := \begin{bmatrix} \beta^k \\ \beta^k_{m_k+1} \end{bmatrix}$.

Case 2. $\mu^k_{max} > \mu$.

Subcase 2.1. There is j_k such that $\mu_{j_k}(z_k) > \mu$ and $\sigma_{j_k}(z_k) < \bar{\sigma}$.

The constraint $(e_{j_k}^\intercal \Lambda^k) z \ge \beta^k_{j_k}$ is deleted. Let $m_{k+1} := m_k - 1$, and Λ^{k+1} and β^{k+1} be Λ^k and β^k respectively with the j_k-th row removed.

Subcase 2.2. For all i such that $\mu_i(z_k) > \mu$, but $\sigma_i(z_k) \geq \bar{\sigma}$, set $K(\alpha_i, \beta_i) := \alpha_i z_k - \beta_i$. Let $m_{k+1} := m_k$, $\Lambda^{k+1} := \Lambda^k$ and $\beta^{k+1} := \beta^k$.

When a constraint is added or deleted the analytic center shifts. A sequence of Newton steps is performed beginning with the current iterate z_k to obtain the next iterate z_{k+1} 'sufficiently close' to the analytic center w_{k+1} of the new polyhedral set. The 'closeness' of z_{k+1} to w_{k+1} is measured in terms of $\psi(z) := \nabla F(z)(\nabla^2 F(z))^{-1} \nabla F(z)$ with $\psi(z_{k+1}) \leq 2\rho^2$ for some small $\rho > 0$. It is proved in [4, §6.2] that $O(1)$ Newton steps suffice to find such an approximation z_{k+1} beginning with z_k.

Our description above leads to the following class of analytic center cutting plane algorithms for (9).

Algorithm Class 3.

Initialization:

```
choose $L > 0$ so that $z^* \in \text{int } \mathcal{P}^0$ where $\mathcal{P}^0$ is as in (38),
$\mu := 2$, $\bar{\sigma} := 0.04$, $\tau := 1/16$;
$z_0 := 0$, $m_0 := 2(n_1 + 1)$, $K(\alpha_i, \beta_i) := 2^L, i = 1, 2, \ldots, m_0$.
```

Main Step:

begin

```
for $k = 0, 1, \ldots$ do
    $\mu^k_{max} := \max_{1 \leq i \leq m_k} \mu_i(z_k)$;
    if $\mu^k_{max} \leq \mu$ then
        $\bar{z}_k := [\bar{x}_k^\mathsf{T}, \theta_k]^\mathsf{T} := P z_k$;
        if $\bar{x}_k \notin S_2$ then
            choose $j_k$ such that $e_{j_k}^\mathsf{T} \bar{x}_k < 0$;
            $\bar{e}_{j_k} := P_A e_{j_k}$;
            $\alpha_k := [\bar{e}_{j_k}^\mathsf{T}, 0]^\mathsf{T}$;
        else
            call the oracle;
            if oracle decides $\bar{x}_k \notin S_3$ then
                $\bar{D}_k := P_A D_k$;
                $\alpha_k := [\bar{D}_k^\mathsf{T}, 0]^\mathsf{T}$;
            else
                if the oracle decides $\mathcal{Q}\bar{x}_k > \bar{\theta}_k$ then
                    $\bar{E}_k := P_A E_k$;
                    $\alpha_k := [\bar{E}_k^\mathsf{T}, 1]^\mathsf{T}$;
                else
                    $\alpha_k := [-\bar{c}^\mathsf{T}, -1]^\mathsf{T}$;
                    end if;
                end if;
            end if;
            define $\Lambda^{k+1}$ and $\beta^{k+1}$ as in Case 1 above to
            add the cut $\alpha_k^\mathsf{T} z \geq \beta^k_{m_k+1}$ and let $m_{k+1} := m_k + 1$;
            take $O(1)$ Newton steps to move to the new approximate
            center $z_{k+1}$;
            Set $K(\alpha_k, \beta^k_{m_k+1}) := \alpha_k z_{k+1} + \beta^k_{m_k+1}$;
        else
            if for some $j_k$, $\mu_{j_k}(z_k) > \mu$ and $\sigma_{j_k}(z_k) < \bar{\sigma}$ then
                $m_{k+1} := m_k - 1$;
                define $\Lambda^{k+1}$ and $\beta^{k+1}$ as in Subcase 2.1 above to delete the
                $j_k$-th cut;
                take $O(1)$ Newton steps to move to the new approximate
                center $z_{k+1}$;
            else
                for all $i$ such that $\mu_i(z_k) > \mu$, but $\sigma_i(z_k) \geq \bar{\sigma}$, set
                $K(\alpha_i, \beta_i) := \alpha_i^\mathsf{T} z_k - \beta_i$, $m_{k+1} := m_k$, $\Lambda^{k+1} := \Lambda^k$ and
                $\beta^{k+1} := \beta^k$;
            end if;
        end for;
    end if;
end
```

Define the following stopping conditions for Algorithm 3 which are justified by Theorem 15 below.

Stopping Condition 1. $m_k \geq \nu(n_1 + 1)L$.

Stopping Condition 2. $\min_{1 \leq i \leq m_k}\{\alpha_i^\mathsf{T} z_k - \beta_i\} \leq 2^{-(L+1)}/(\nu(n_1+1)L)$, where $\nu > 0$ is a suitable constant. Following [4, Theorem 14], $\nu = 2700$ would suffice.

In addition define the sequence $\{\hat{Z}_k\}$ by

$$\hat{Z}_{k+1} := \begin{cases} -\beta^k_{m_k+1} & \text{if an objective cut } \alpha_k^\mathsf{T} z \geq \beta^k_{m_k+1} \\ & \text{with } -\beta^k_{m_k+1} < \hat{Z}_k \text{ is added} \\ \hat{Z}_k & \text{otherwise,} \end{cases} \tag{40}$$

where $\hat{Z}_0 := \infty$ for $k = 0, 1, \ldots$.

The following result follows from [4] and shows that one of the two stopping conditions must be met in $O(n_1 L^2)$ iterations.

Theorem 15 *Suppose that the set $S_\epsilon \cap \mathcal{P}^0$ contains a ball of radius 2^{-L} where S_ϵ is defined by (17). Suppose further that we terminate Algorithm 3 at itera-*

tion k if Stopping Condition 1 or 2 is satisfied. Then Algorithm 3 terminates with Stopping Condition 1 in $O(n_1 L)$ iterations or with Stopping Condition 2 in $O(n_1 L^2)$ iterations. The termination of Algorithm 3 at iteration k by either stopping condition implies that the upper bound $\hat{Z}_k$ satisfies $\hat{Z}_k \leq c^\mathsf{T} x^* + \theta^* + \epsilon$, which further implies that $[\hat{x}_k^\mathsf{T}, \hat{\theta}_k]^\mathsf{T} \in S_\epsilon$ where $[\hat{x}_k^\mathsf{T}, \hat{\theta}_k]^\mathsf{T}$ is defined by (40) and (19). The complexity of the algorithm is $O(Tn_1 L^2 + n_1^4 L^3)$ arithmetic operations where T is the cost of a call to the oracle in arithmetic operations.

PROOF. The conclusion on the number of iterations follows from [4, Theorem 14] and [4, Theorem 15]. Since the number of arithmetic operations per iteration is $O(T + n_1^3)$ for the calls to the oracle and for the $O(1)$ Newton steps, the conclusion on the overall complexity of Algorithm 3 follows. □

As in the case of Algorithm 2 using fast multiplications the complexity of Algorithm 3 can be reduced to $O(Tn_1 L^2 + n_1 M(n_1) L^3)$ where $M(n_1) = n_1^{2.38}$.

5 The work of the oracle

In this section we indicate how the work of the oracle which Algorithms 1, 2 and 3 call may be performed.

Recall from the discussion at the end of §1 that the oracle is always presented with $\bar{z} := [\bar{x}^\mathsf{T}, \bar{\theta}]^\mathsf{T}$ with $\bar{x} \in S_1 \cap S_2$. The oracle is expected to do the following.

(a) Decide whether $\bar{x} \in S_3 := \{x \in \mathbb{R}^{n_1} : (\exists y \in \mathbb{R}^{n_2} : My = h - Tx, y \geq 0)\}$ or not, returning $D \in \mathbb{R}^{n_1}$ and $d \in \mathbb{R}$ such that $D^\mathsf{T} x \geq d$ for all $x \in S_3$ and $D^\mathsf{T} \bar{x} < d$ if $\bar{x} \notin S_3$.

(b) If $\bar{x} \in S_3$ decide whether $\mathcal{Q}(\bar{x}) := \min_{y \in \mathbb{R}^{n_2}} \{q^\mathsf{T} y : My = h - T\bar{x}, y \geq 0\} \leq \bar{\theta}$ or not, returning $E \in \mathbb{R}^{n_1}$ and $e \in \mathbb{R}$ such that $E^\mathsf{T} x + \theta \geq e$ for all $x \in S_1 \cap S_2 \cap S_3$ and all θ with $\mathcal{Q}(x) \leq \theta$, and $E^\mathsf{T} \bar{x} + \bar{\theta} < e$, if indeed $\mathcal{Q}(\bar{x}) \geq \bar{\theta}$.

(c) In the case of Algorithm 1, if deep cuts are desired, then when $\bar{x} \in S_1 \cap S_2 \cap S_3$ and $\mathcal{Q}(\bar{x}) \leq \bar{\theta}$, return $\theta' \leq \bar{\theta}$ where $\theta' := q^\mathsf{T} y$ with y satisfying $My = h - T\bar{x}$ and $y \geq 0$.

Now define the "phase-I" problem

$$\begin{array}{rl} \text{minimize} & e^\mathsf{T} v_+ + e^\mathsf{T} v_- \\ \text{subject to} & My + v_+ - v_- = h - T\bar{x} \\ & y, \quad v_+, \quad v_- \geq 0 \end{array} \tag{41}$$

associated with the lp

$$\begin{aligned} \text{minimize} \quad & q^{\mathsf{T}} y \\ \text{subject to} \quad & My = h - T\bar{x} \\ & y \geq 0 \end{aligned} \tag{42}$$

whose optimal value defines $\mathcal{Q}$. The duals of (41) and (42) respectively are

$$\begin{aligned} \text{maximize} \quad & (h - T\bar{x})^{\mathsf{T}} w \\ \text{subject to} \quad & M^{\mathsf{T}} w \leq 0 \\ & -e \leq w \leq e \end{aligned} \tag{43}$$

and

$$\begin{aligned} \text{maximize} \quad & (h - T\bar{x})^{\mathsf{T}} w \\ \text{subject to} \quad & M^{\mathsf{T}} w \leq q \end{aligned} \tag{44}$$

respectively.

It is clear that any feasible solution w for (43) with

$$(h - T\bar{x})^{\mathsf{T}} w > 0 \tag{45}$$

can be used to generate the $D \in \mathbb{R}^{n_1}$, $d \in \mathbb{R}$ in (a) of the oracle work with $D := T^{\mathsf{T}} w$ and $d := h^{\mathsf{T}} w$ since for such D and d, $D^{\mathsf{T}} x \geq d$ for all $x \in S_3$ and $D^{\mathsf{T}} \bar{x} < d$.

Any feasible solution w to (44) which satisfies

$$(h - T\bar{x})^{\mathsf{T}} w > \bar{\theta} \tag{46}$$

can be used to generate E and e in (b) of the oracle work with $E := T^{\mathsf{T}} w$ and $e := h^{\mathsf{T}} w$ which separates the current iterate $\bar{z}$ and is satisfied by all z feasible to (9).

Also any feasible solution y to (42) which satisfies

$$q^{\mathsf{T}} y \leq \bar{\theta} \tag{47}$$

gives θ' of (c) of the oracle work with $\theta' := q^{\mathsf{T}} y$.

Note that by the above analysis the oracle work does not necessarily involve solution of lp's in contrast to the case of simplex-based algorithms such as the Van Slyke and Wets algorithm [40].

According to the above analysis one easy way to perform the oracle work is to solve feasibility problems as follows. Attempt to find a feasible solution to the system

$$\begin{aligned} M^{\mathsf{T}} w &\leq 0 \\ -e \leq w &\leq e \\ (h - T\bar{x})^{\mathsf{T}} w &\geq 0. \end{aligned} \tag{48}$$

If a feasible solution w to (48) with $(h - T\bar{x})^{\mathsf{T}} w > 0$ is found, then we conclude that $\bar{x} \notin S_3$, set $D := T^{\mathsf{T}} w$ and $d := h^{\mathsf{T}} w$, and terminate the oracle work. Otherwise, $\bar{x} \in S_3$, and we attempt to find a feasible solution to the system

$$\begin{aligned} M^{\mathsf{T}} w &\leq q \\ (h - T\bar{x})^{\mathsf{T}} w &\geq \bar{\theta}. \end{aligned} \tag{49}$$

If a feasible solution w to (49) with $(h - T\bar{x})^{\mathsf{T}} w > \bar{\theta}$ is found, then we conclude that $\mathcal{Q}(\bar{x}) > \bar{\theta}$, set $E := T^{\mathsf{T}} w$ and $e := h^{\mathsf{T}} w$ and terminate the oracle work. Otherwise, $\mathcal{Q}(\bar{x}) \leq \bar{\theta}$ and we can simply use $\theta' := \bar{\theta}$. (In the case of Algorithm 1 we would of course have a central cut in Subcase 2.2.2.).

We can use the ellipsoid algorithm on the two feasibility problems (48) and (49). It is reasonable to make the following assumption.

(A3) L in Algorithms 1, 2 and 3 is such that the set of all w satisfying (48) is contained in a ball of radius 2^L centered at origin, and contains a ball of radius 2^{-L} with $\bar{x} := \bar{x}_k$ for all $\bar{x}_k$ generated by Algorithm 1, 2 or 3. Moreover, L is such that the set of all w satisfying (49) is contained in a ball of radius 2^L centered at the origin, and if that set is nonempty it contains a ball of radius 2^{-L} with $\bar{x} := \bar{x}_k$ for all $\bar{x}_k$ generated by Algorithm 1, 2 or 3.

If (A3) holds, then the oracle work performed by using the ellipsoid algorithm on the feasibility problems (48) and (49) would have complexity $O((n_2 m_2^3 + m_2^4)L)$ in terms of arithmetic operations. So $T = O((n_2 m_2^3 + m_2^4)L)$.

We can also use the volumetric center algorithm [2,37] on the feasibility problems (48) and (49). Then under Assumption (A3) $T = O(m_2^{3.38} L)$.

Alternatively, we can use the analytic center algorithm of [4]. Then under the analog of Assumption (A3) $T = O(m_2^{3.38} L^3)$.

It is possible to perform the oracle work using linear programming algorithms, such as the ellipsoid algorithm [12], the algorithm of Vaidya [39] or the algorithm of Renegar [34]. Consider the use of the algorithm of Vaidya [39]. Apply the algorithm on (43) and at each iterate w check whether (45) is satisfied. If so, the required D and d are given by $D := T^{\mathsf{T}}w$ and $d := h^{\mathsf{T}}w$. Otherwise, the algorithm of Vaidya [39] terminates without satisfying (45). Then apply the algorithm on (44) and at each iterate w check whether (46) is satisfied. If so, the required E and e are given by $E := T^{\mathsf{T}}w$ and $e := h^{\mathsf{T}}w$. Otherwise, the algorithm will terminate with $\mathcal{Q}(\bar{x})$ and we can use $\theta' := \mathcal{Q}(\bar{x})$. It is reasonable to make the following assumption.

(A4) L in Algorithm 1, 2 and 3 is such that the feasible sets of (43) and (44) are contained in a ball of radius 2^L centered at origin and that each of these sets contains a ball of radius 2^{-L}.

Under Assumption (A4), the analysis in [39] gives that the oracle work $T = O((n_2 + m_2)m_2^2 + (n_2 + m_2)^{1.5}m_2)L)$ in terms of arithmetic operations.

Similarly, if the ellipsoid algorithm is used, under (A4), $T = ((n_2 m_2^3 + m_2^4)L)$.

Another interesting way to perform the oracle work is to use the primal-dual path following algorithm as follows. Consider the use of the algorithm of Monteiro and Adler [32]. Apply the algorithm on (41) and its dual (43), and at each iterate (y, w) check whether (45) is satisfied. If so, the required D and d are given by $D := T^{\mathsf{T}}w$ and $d := h^{\mathsf{T}}w$. Otherwise, the algorithm terminates without satisfying (45). Then apply the algorithm on (42) and its dual (44), and at each iterate (y, w) check whether (46) or (47) is satisfied. If (46) is satisfied, the required E and e are given by $E := T^{\mathsf{T}}w$ and $e := h^{\mathsf{T}}w$. If (47) is satisfied, the required θ' is given by $\theta' := q^{\mathsf{T}}y$. The the analysis in [32] gives that the oracle work $T = O(m_2 n_2^2 L)$ in terms of arithmetic operations.

6 Application to stochastic linear programs

In this section we consider the general case of problem (1,2,3) with $K > 1$. As in the case of $K := 1$, we do not lose generality by assuming that $b = 0$. We will further assume that A has full row rank.

Define

$$S_3^l := \{x \in \mathbb{R}^{n_1} : (\exists y \in \mathbb{R}^{n_2} : M^l y = h^l - T^l x, y \geq 0)\} \text{ for } l = 1, 2, \ldots, K$$

and $S_3 := \cap_{l=1}^{K} S_3^l$. Then problem (1,2,3) is equivalent to

$$\begin{aligned} \text{minimize} \quad & Z(x) := c^\mathsf{T} x + \theta \\ \text{subject to} \quad & \mathcal{Q}(x) \leq \theta \\ & \qquad x \in S := S_1 \cap S_2 \cap S_3 \end{aligned} \tag{50}$$

which is symbolically the same as (9) except that $\mathcal{Q}$ is defined as in (1,2,3).

We assume that (50) has a minimizer $[x^{*\mathsf{T}}, \theta^*]^\mathsf{T}$.

The similarity of problems (50) and (9) suggests that Algorithm 1, 2 and 3 we stated for (9) could be modified for (50). This indeed is the case. We now state the analogs of Algorithms 1, 2 and 3 for (50). In the analog of Algorithm 1 we assume that oracle work is performed using the ellipsoid algorithm for appropriate lp's in a manner analogous to that described in §5 for problem (9). In the other two algorithms we assume that oracle work is performed using the linear programming algorithm of Vaidya [39] analogous to the way outlined in §5 for problem (9). Of course other ways of performing the oracle work can also be used to obtain different variants of algorithms we present below. Following the statement of each algorithm we also present a theorem on its complexity. Proofs of these theorems follow from the results in §§2, 3 and 4 and our discussion of the work of the oracle in §5.

In the algorithms below we need to refer to the lp's

$$\begin{aligned} \text{maximize} \quad & (h^l - T^l \bar{x})^\mathsf{T} w \\ \text{subject to} \quad & (M^l)^\mathsf{T} w \leq 0 \\ & -e \leq w \leq e, \end{aligned} \tag{51}$$

and

$$\begin{aligned} \text{maximize} \quad & (h^l - T^l \bar{x})^\mathsf{T} w \\ \text{subject to} \quad & (M^l)^\mathsf{T} w \leq q^l. \end{aligned} \tag{52}$$

Algorithm 4 (Ellipsoid algorithm for problem (50)).

Initialization:

choose $x_0 \in \mathbb{R}^{n_1}$, $\theta_0 \in \mathbb{R}$ and $L > 0$ such that with $z_0 := [x_0^\mathsf{T}, \theta_0]^\mathsf{T}$ and $B_0 := 2^{2L} I \in \mathbb{R}^{(n_1+1)\times(n_1+1)}$,
$z^* \in \operatorname{int}(\mathcal{E}_0 := \{z \in \mathbb{R}^{n_1+1} : (z - z_0)^\mathsf{T} B_0^{-1} (z - z_0) \leq 1\})$.

Main Step:

```
begin
  for $k = 0, 1, \ldots$ do
    $\bar{z}_k := [\bar{x}_k^\mathsf{T}, \theta_k]^\mathsf{T} := P z_k$;
    if $\bar{x}_k \notin S_2$ then
      choose $j_k$ such that $e_{j_k}^\mathsf{T} \bar{x}_k < 0$;
      $\bar{e}_{j_k} := P_A e_{j_k}$;
      $\alpha_k := [-\bar{e}_{j_k}^\mathsf{T}, 0]^\mathsf{T}$; $c_k := 0$;
    else
      begin (the oracle work);
        for $l = 1, 2, \ldots, K$ in parallel do
          apply ellipsoid algorithm on (51) to generate $\{w_{j_l}^l\}$;
          if $w_{j_l}^l$ is feasible for (51) and $(h^l - T^l \bar{x}_k)^\mathsf{T} w_{j_l}^l > 0$
          for some $l$, $j_l$ then
            $D_k := (T^l)^\mathsf{T} w_{j_l}^l$, $d_k := (h^l)^\mathsf{T} w_{j_l}^l$;
            $\bar{D}_k := P_A D_k$;
            $\alpha_k := [-\bar{D}_k^\mathsf{T}, 0]^\mathsf{T}$; $c_k := -d_k$;
            exit parallel do;
          end if
        end parallel do
        for $l = 1, 2, \ldots, K$ in parallel do
          apply ellipsoid algorithm on (52) to generate $\{w_{j_l}^l\}$;
          if $w_{j_l}^l$ is feasible for (52) and $\sum_{l=1}^K p^l (h^l - T^l \bar{x}_k)^\mathsf{T} w_{j_l}^l > \bar{\theta}_k$
          for any set of indices $j_l$ and $l = 1, 2, \ldots K$ then
            $E_k := \sum_{l=1}^K p^l (T^l)^\mathsf{T} w_{j_l}^l$; $e_k := \sum_{l=1}^K p^l (h^l)^\mathsf{T} w_{j_l}^l$;
            $\bar{E}_k := P_A E_k$;
            $\alpha_k := [-\bar{E}_k^\mathsf{T}, -1]^\mathsf{T}$; $c_k := -e_k$;
            exit parallel do;
          end if
          $w_*^l :=$ terminating member of $\{w_{j_l}^l\}$;
        end parallel do
        $\theta_k' := \sum_{l=1}^K p^l (h^l - T^\mathsf{T} \bar{x}_k)^\mathsf{T} w_*^l$;
        $Z_k := c^\mathsf{T} \bar{x}_k + \theta'$;
        $\alpha_k := [\bar{c}^\mathsf{T}, 1]^\mathsf{T}$; $c_k := Z_k$;
      end (the oracle work);
    end if;
    update $z_k$ and $B_k$ to $z_{k+1}$ and $B_{k+1}$ respectively using (14,15);
  end for;
end
```

Theorem 16 *Suppose that the set $S_\epsilon \cap \mathcal{E}_0$ contains a ball of radius 2^{-L} and that Assumption (A4) holds with (43) and (44) replaced by (51) and (52) for $l = 1, 2, \ldots, K$. Let $N := 4(n_1 + 1)(n_1 + 2) L \ln 2$. Let the sequence $\{[\bar{x}_k^\mathsf{T}, \theta_k]^\mathsf{T}\}$ be generated by Algorithm 4 and let the sequence $\{[\hat{x}_k^\mathsf{T}, \hat{\theta}_k]^\mathsf{T}\}$ be defined by (18,19). Then, $[\hat{x}_k^\mathsf{T}, \hat{\theta}_k]^\mathsf{T} \in S_\epsilon$ for all $k \geq N$. The complexity of the algorithm* *is* $O(Tn_1^2L + n_1^4L)$ *arithmetic operations where* $T := O((n_2m_2^3 + m_2^4)L)$.

We mention in passing that it is possible to analyze Algorithm 4 under weaker hypotheses using techniques in [18].

Algorithm 5 (Volumetric center algorithm for problem (50)).

Initialization:

```
    choose $L > 0$ so that $z^* \in \text{int } \mathcal{P}^0$ where $\mathcal{P}^0$ is as in (22),
    $\bar{\sigma} \in (0, 1)$, $\tau > 0$, $\tau_1 > 0$;
    define $z_0$ by (23).
```

Main Step:

```
begin

    for $k = 0, 1, \ldots$ do
        $\sigma_{min}^k := \sigma_{min}(z_k)$;
        if $\sigma_{min}^k \geq \bar{\sigma}$ then
            $\bar{z}_k := [\bar{x}_k^\mathsf{T}, \theta_k]^\mathsf{T} := Pz_k$;
            if $\bar{x}_k \notin S_2$ then
                choose $j_k$ such that $e_{j_k}^\mathsf{T}\bar{x}_k < 0$;
                $\bar{e}_{j_k} := P_A e_{j_k}$;
                $\alpha_k := [\bar{e}_{j_k}^\mathsf{T}, 0]^\mathsf{T}$;
            else
                begin (the oracle work);
                    for $l = 1, 2, \ldots, K$ in parallel do
                        apply Vaidya's algorithm [39] on (51) to generate $\{w_{j_l}^l\}$;
                        if $w_{j_l}^l$ is feasible for (51) and $(h^l - T^l\bar{x}_k)^\mathsf{T} w_{j_l}^l > 0$
                        for some $l$, $j_l$ then
                            $D_k := (T^l)^\mathsf{T} w_{j_l}^l$;
                            $\bar{D}_k := P_A D_k$;
                            $\alpha_k := [\bar{D}_k^\mathsf{T}, 0]^\mathsf{T}$;
                            exit parallel do;
                        end if
                    end parallel do;
                    for $l = 1, 2, \ldots, K$ in parallel do
                        apply Vaidya's algorithm [39] on (52) to generate $\{w_{j_l}^l\}$;
                        if $w_{j_l}^l$ is feasible for (52) and $\sum_{l=1}^K p^l(h^l - T^l\bar{x}_k)^\mathsf{T} w_{j_l}^l > \bar{\theta}_k$
                        for any set of indices $j_l$ and $l = 1, 2, \ldots, K$ then
                            $E_k := \sum_{l=1}^K p^l (T^l)^\mathsf{T} w_{j_l}^l$;
                            $\bar{E}_k := P_A E_k$;
                            $\alpha_k := [\bar{E}_k^\mathsf{T}, 1]^\mathsf{T}$;
                            exit parallel do;
                        end if
                    end parallel do;
                    $\alpha_k := [-\bar{c}^{\mathsf{T}}, -1]^{\mathsf{T}}$;
                end (the oracle work);
            end if;
            define $\Lambda^{k+1}$ and $\beta^{k+1}$ as in Case 1 above to
            add the cut (25) and let $m_{k+1} := m_k + 1$
            or translate the cut (27) and let $m_{k+1} := m_k$;
          else
            $m_{k+1} := m_k - 1$;
            define $\Lambda^{k+1}$ and $\beta^{k+1}$ as in Case 2 above to
            delete the cut (28);
          end if;
          beginning with $z_k$, take a sequence of pure Newton steps to obtain
          $z_{k+1}$ 'close' to new volumetric center;
        end for;
end
```

Theorem 17 *Suppose that the set $S_\epsilon \cap \mathcal{P}_0$ contains a ball of radius 2^{-L} and that Assumption (A4) holds with (43) and (44) replaced by (51) and (52) for $l = 1, 2, \ldots, K$. Let the sequence $\{[\bar{x}_k^{\mathsf{T}}, \theta_k]^{\mathsf{T}}\}$ be generated by Algorithm 5 and let the sequence $\{[\hat{x}_k^{\mathsf{T}}, \hat{\theta}_k]^{\mathsf{T}}\}$ be defined by (18,19). Then, $[\hat{x}_k^{\mathsf{T}}, \hat{\theta}_k]^{\mathsf{T}} \in S_\epsilon$ for all $k \geq N_{vol} = O(n_1 L)$ where N_{vol} is defined in Theorem 14. The complexity of the algorithm is $O(Tn_1L + n_1^4L)$ arithmetic operations where $T = O(((n_2 + m_2)m_2^2 + (n_2 + m_2)^{1.5}m_2)K)$. If fast matrix multiplication [13] is used the complexity of the algorithm is $O(Tn_1L + n_1M(n_1)L)$ arithmetic operations where $M(n_1) = n_1^{2.38}$.*

Algorithm 6 (Analytic center algorithm for problem (50)).

Initialization:

```
  choose $L > 0$ so that $z^* \in \text{int } \mathcal{P}^0$ where $\mathcal{P}^0$ is as in (38),
  $\mu := 2$, $\bar{\sigma} := 0.04$, $\tau := 1/16$;
  $z_0 := 0$, $m_0 := 2(n_1 + 1)$, $K(\alpha_i, \beta_i) := 2^L, i = 1, 2, \ldots, m_0$.
```

Main Step:

```
begin

  for $k = 0, 1, \ldots$ do
    $\mu_{max}^k := \max_{1 \leq i \leq m_k} \mu_i(z_k)$;
    if $\mu_{max}^k \leq \mu$ then
      $\bar{z}_k := [\bar{x}_k^{\mathsf{T}}, \theta_k]^{\mathsf{T}} := Pz_k$;
      if $\bar{x}_k \notin S_2$ then
```

```
            choose $j_k$ such that $e_{j_k}^{\mathsf{T}}\bar{x}_k < 0$;
            $\bar{e}_{j_k} := P_A e_{j_k}$;
            $\alpha_k := [\bar{e}_{j_k}^{\mathsf{T}}, 0]^{\mathsf{T}}$;
        else
            begin (the oracle work)
                for $l = 1, 2, \ldots, K$ in parallel do
                    apply Vaidya's algorithm [39] on (51) to generate $\{w_{j_l}^l\}$;
                    if $w_{j_l}^l$ is feasible for (51)and $(h^l - T^l\bar{x}_k)^{\mathsf{T}} w_{j_l}^l > 0$
                    for some $l$, $j_l$ then
                        $D_k := (T^l)^{\mathsf{T}} w_{j_l}^l$;
                        $\bar{D}_k := P_A D_k$;
                        $\alpha_k := [\bar{D}_k^{\mathsf{T}}, 0]^{\mathsf{T}}$;
                        exit parallel do;
                    end if
                end parallel do;
                for $l = 1, 2, \ldots, K$ in parallel do
                    apply Vaidya's algorithm [39] on (52) to generate $\{w_{j_l}^l\}$;
                    if $w_{j_l}^l$ is feasible for (52) and $\sum_{l=1}^K p^l (h^l - T^l\bar{x}_k)^{\mathsf{T}} w_{j_l}^l > \bar{\theta}_k$
                    for any set of indices $j_l$ and $l = 1, 2, \ldots, K$ then
                        $E_k := \sum_{l=1}^K p^l (T^l)^{\mathsf{T}} w_{j_l}^l$;
                        $\bar{E}_k := P_A E_k$;
                        $\alpha_k := [\bar{E}_k^{\mathsf{T}}, 1]^{\mathsf{T}}$;
                        exit parallel do;
                    end if
                end parallel do;
                $\alpha_k := [-\bar{c}^{\mathsf{T}}, -1]^{\mathsf{T}}$;
            end(the oracle work);
        end if;
        define $\Lambda^{k+1}$ and $\beta^{k+1}$ as in Case 1 above to
        add the cut $\alpha_k^{\mathsf{T}} z \geq \beta_{m_k+1}^k$ and let $m_{k+1} := m_k + 1$;
        take $O(1)$ Newton steps to move to the new approximate
        center $z_{k+1}$;
        Set $K(\alpha_k, \beta_{m_k+1}^k) := \alpha_k z_{k+1} + \beta_{m_k+1}^k$;
    else
        if for some $j_k$, $\mu_{j_k}(z_k) > \mu$ and $\sigma_{j_k}(z_k) < \bar{\sigma}$ then
            $m_{k+1} := m_k - 1$;
            define $\Lambda^{k+1}$ and $\beta^{k+1}$ as in Subcase 2.1 above to delete the
            $j_k$-th cut;
            take $O(1)$ Newton steps to move to the new approximate
            center $z_{k+1}$;
        else
            for all $i$ such that $\mu_i(z_k) > \mu$, but $\sigma_i(z_k) \geq \bar{\sigma}$, set
            $K(\alpha_i, \beta_i) := \alpha_i^{\mathsf{T}} z_k - \beta_i$, $m_{k+1} := m_k$, $\Lambda^{k+1} := \Lambda^k$
            and $\beta^{k+1} := \beta^k$;
```

```
                end if
            end if
        end for;
    end
```

Theorem 18 *Suppose that the set $S_\epsilon \cap \mathcal{P}^0$ contains a ball of radius 2^{-L} and that Assumption (A4) holds with (43) and (44) replaced by (51) and (52) for $l = 1, 2, \ldots, K$. Suppose further that we terminate Algorithm 6 at iteration k if Stopping Condition 1 or 2 defined in §4 for Algorithm 3 is satisfied. Then Algorithm 6 terminates with Stopping Condition 1 in $O(n_1 L)$ iterations or with Stopping Condition 2 in $O(n_1 L^2)$ iterations. The termination of Algorithm 6 at iteration k by either stopping condition implies that the upper bound $\hat{Z}_k$ defined by (40) satisfies $\hat{Z}_k \le c^\top x^* + \theta^* + \epsilon$, which further implies that $[\hat{x}_k^\top, \hat{\theta}_k]^\top \in S_\epsilon$ where $[\hat{x}_k^\top, \hat{\theta}_k]^\top$ is defined by (40) and (19). The complexity of the algorithm is $O(Tn_1L^2 + n_1^4 L^3)$ arithmetic operations where $T = O(((n_2 + m_2)m_2^2 + (n_2 + m_2)^{1.5} m_2)KL)$. If fast matrix multiplications are used the complexity of the algorithm is $O(Tn_1L^2 + n_1^{3.38} L^3)$.*

We have included parallel do loops in the statements of Algorithms 4, 5 and 6 to indicate the potential for using parallel processors in the implementation of these algorithms. In this paper we have done this in the most obvious manner, and note that in particular, the important case in which K is bigger than the available number of processors needs more careful consideration.

We conclude the paper by restating the complexity bounds obtained above for Algorithms 4, 5 and 6 for two-stage stochastic programs together with the complexity bound obtained by Birge and Qi [11] for their algorithm for easy comparison. In particular, note that Algorithms 4, 5 and 6 have complexities that are linear in the number of realizations K.

Complexity of algorithm of Birge and Qi [11]:
$O((n^{0.5} n_2^2 + n \max\{n_1, n_2\} + n_1^3) nL)$, $n := n_1 + Kn_2$.

Complexity of Algorithm 4 (ellipsoid):
$O(Tn_1^2 L + n_1^4 L)$, $T := O(((n_2 + m_2)m_2^2 + (n_2 + m_2)^{1.5} m_2)K)$

Complexity of Algorithm 5 (volumetric center):
$O(Tn_1 L + n_1^{3.38} L)$, $T := O(((n_2 + m_2)m_2^2 + (n_2 + m_2)^{1.5} m_2)K)$

Complexity of Algorithm 6 (analytic center):
$O(Tn_1 L^2 + n_1^{3.38} L^3)$, $T := O(((n_2 + m_2)m_2^2 + (n_2 + m_2)^{1.5} m_2)K)$

References

[1] K. M. Anstreicher, Large step volumetric potential reduction algorithms for linear programming, Technical Report, Department of Management Sciences, University of Iowa (Iowa City, IA, 1994).

[2] K. M. Anstreicher, On Vaidya's volumetric cutting plane method for convex programming, Department of Management Sciences, University of Iowa (Iowa City, IA 1994).

[3] K. A. Ariyawansa and D. D. Hudson, Performance of a benchmark implementation of the Van Slyke and Wets algorithm for stochastic two-stage stochastic programs on the Sequent/Balance, *Concurrency: Practice and Experience* **3** (1991) 109–128.

[4] D. S. Atkinson and P. M. Vaidya, A cutting plane algorithm for convex programming that uses analytic centers, *Mathematical Programming* **69** (1995) 1–43.

[5] O. Bahn, O. du Merle, J.-L. Goffin and J.-P. Vial, Experimental behavior of an interior point cutting plane algorithm for convex programming, *Discrete Applied Mathematics* **49** (1994) 3–23.

[6] O. Bahn, O. du Merle, J.-L. Goffin and J.-P. Vial, A cutting plane method from analytic centers for stochastic programming, *Mathematical Programming* **69** (1995) 45–73.

[7] J. F. Benders, Partitioning procedures for solving mixed-variables programming problems, *Numerische Mathematik* **4** (1962) 238–252.

[8] D. Bertsimas and J. B. Orlin, A technique for speeding up the solution of the Lagrangean dual, *Mathematical Programming* **63** (1994) 23–45.

[9] J. R. Birge, An L-shaped method computer code for multi-stage stochastic linear programs, in: Y. Ermoliev and R. J.-B. Wets, eds., *Numerical Techniques for Stochastic Optimization* (Springer-Verlag, Berlin, 1988) 255–266.

[10] J. R. Birge and D. F. Holmes, Efficient solution of two-stage stochastic linear programs using interior point methods, *Computational Optimization and Applications* **1** (1992) 245–276.

[11] J. R. Birge and L. Qi, Computing block-angular Karmarkar projections with applications to stochastic programming, *Management Science* **34** (1988) 1472–1479.

[12] R. G. Bland, D. Goldfarb and M. J. Todd, The ellipsoid method: a survey, *Operations Research* **29** (1981) 1039–1091.

[13] D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progressions, *Preceedings of the 19th Annual ACM Symposium on Theory of Computing* (1987) 1–6.

[14] J. Czyzyk, R. Fourer and S. Mehrotra, A study of the augmented system and column-splitting approaches for solving two-stage stochastic linear programs by interior point methods, *ORSA Journal on Computing* (1995) 474–490.

[15] G. B. Dantzig and P. Wolfe, The decomposition algorithm for linear programming, *Econometrica* **29** (1961) 425–453.

[16] G. B. Dantzig and A. Madansky, On the solution of two-stage linear programs under uncertainty, in: *Proceedings of Fourth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1* (University of California Press, Berkeley, CA, 1961) 165–176.

[17] G. B. Dantzig and Y. Ye, A build-up interior method for linear programming: Affine scaling form, Technical Report SOL 90-4, Systems Optimization Laboratory, Department of Operations Research, Stanford University, (Stanford, CA, 1990).

[18] J. B. G. Frenk, J. Gromicho and S. Zhang, A deep cut ellipsoid algorithm for convex programming: Theory and applications, *Mathematical Programming* **63** (1994) 83–108.

[19] D. den Hertog, C. Roos and T. Terlaky, A large-step analytic center method for a class of smooth convex programming problems, *SIAM J. Optim.* **2** (1992) 55–70.

[20] D. den Hertog, C. Roos and T. Terlaky, A build-up variant of the logarithmic barrier method for LP, *OR Letters* **12** (1992) 181–186.

[21] D. den Hertog, C. Roos and T. Terlaky, A complexity reduction for the long-step path following algorithm for linear programming, vvv*SIAM J. Optim.* **2** (1992) 71–87.

[22] H. I. Gassmann, MSLiP: a computer code for the multistage stochastic linear programming problem, *Mathematical Programming* **47** (1990) 407–423.

[23] J.-L. Goffin and J.-P. Vial, Cutting planes and column generation techniques with the projective algorithm, *SIAM J. Optim.* **1** (1990) 409-429.

[24] J.-L. Goffin, A, Haurie and J.-P. Vial, Decomposition and nondifferentiable optimization with the projective algorithm, *Managment Science* **38** (1992) 284–302.

[25] M. Grötschel, L. Lovasz and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization (Second Corrected Edition)* (Springer-Verlag, Berlin, 1991).

[26] E. Jessup, D. Yang and S. A. Zenios, Parallel factorization of structured matrices arising in stochastic programming, Technical Report 93-02, Department of Public and Business Administration, University of Cyprus (Nicosia, Cyprus, 1993).

[27] P. L. Jiang, Polynomial cutting plane algorithms for stochastic programming and related problems, Ph. D. dissertation, Department of Pure and Applied Mathematics, Washington State University (Pullman, Washington, 1997).

[28] P. Kall, Computational methods for solving two-stage stochastic linear programming problems, *Z. Angew. Math. Phys.* **30** (1979) 261–271.

[29] L. Khachiyan, A polynomial algorithm in linear programming, *Doklady Akademiia Nauk SSSR* **244** (1979) 1093–1096 [Translated in *Soviet Mathematics Doklady* **20** (1979) 191–194.]

[30] I. J. Lustig, J. M. Mulvey and T. J. Carpenter, Formulating two-stage stochastic programs for interior point methods, *Operations Research* **39** (1991) 757–770.

[31] J. E. Mitchell and M. J. Todd, Solving combinatorial optimization problems using Karmarkar's algorithm, *Mathematical Programming* **56** (1992) 245–284.

[32] R. C. Monteiro and I. Adler, Interior path-following primal-dual algorithms, Part I: Linear programming, *Mathematical Programming* **44** (1989) 27–41.

[33] Y. Nesterov, Complexity estimates of some cutting plane methods based on the analytic barrier, *Mathematical Programming* **69** (1995) 149–176.

[34] J. Renegar, A polynomial-time algorithm based on Newton method for linear programming, *Mathematics Programming* **40** (1988) 59–93.

[35] G. Sonnevend, An analytical centre for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming, in: *Lecture Notes Control Inform. Sci 84* (Springer-Verlag, New York, NY, 1985) 866–876.

[36] B. Strazicky, Some results concerning an algorithm for the discrete recourse problem, in: M. A. H. Dempster, ed., *Stochastic Programming* (Academic Press, London, 1980) 263–274.

[37] P. M. Vaidya, A new algorithm for minimizing convex functions over convex sets, Technical Report, AT&T Bell Laboratories, (Murray Hill, NJ, 1989).

[38] P. M. Vaidya, A new algorithm for minimizing convex functions over convex sets, in: *Annual Symposium on Foundations of Computer Science, Research Triangle Park, NC, 1989* (IEEE Computer Society Press, Los Alamitos, CA 1990) 338–343.

[39] P. M. Vaidya, An algorithm for linear programming which requires $O(((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations, *Mathematical Programming* **47** (1990) 175–201.

[40] R. M. Van Slyke and R. J.-B. Wets, L-shaped linear programs with applications to optimal control and stochastic programming, *SIAM J. Appl. Math.* **17** (1969) 638–663.

[41] R. J.-B. Wets, Stochastic programming: solution techniques and approximation schemes, in: A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming—The State of the Art, Bonn 1982* (Springer-Verlag, Berlin, 1983) 566–603.

[42] Y. Ye, The "build-down" scheme for linear programming, *Mathematical Programming* **46** (1990) 61–72.

[43] Y. Ye, A potential reduction algorithm allowing column generation, *SIAM J. Optim.* **2** (1992) 7–20.