Optimization Methods and Software

ISSN: 1055-6788 (Print) 1029-4937 (Online) Journal homepage: http://www.tandfonline.com/loi/goms20

An improved version of Chubanov's method for solving a homogeneous feasibility problem

Kees Roos

To cite this article: Kees Roos (2018) An improved version of Chubanov's method for solving a homogeneous feasibility problem, Optimization Methods and Software, 33:1, 26-44, DOI: 10.1080/10556788.2017.1368509

To link to this article: https://doi.org/10.1080/10556788.2017.1368509

Optimization Methods & Software, 2018
Vol. 33, No. 1, 26–44, https://doi.org/10.1080/10556788.2017.1368509

An improved version of Chubanov's method for solving a homogeneous feasibility problem

Kees Roos*

Faculty of Information Technology and Systems, Delft University of Technology, Delft, Netherlands

(*Received 6 January 2017; accepted 9 August 2017*)

We deal with a recently proposed method of Chubanov [*A polynomial projection algorithm for linear feasibility problems*. Math. Program. 153 (2015), pp. 687–713] for solving linear homogeneous systems with positive variables. Some improvements of Chubanov's method and its analysis are presented. We propose a new and simple cut criterion and show that the cuts defined by the new criterion are at least as sharp as in [1]. The new cut criterion reduces the iteration bound for his Basic Procedure by a factor 5, without changing the order of its strongly polynomial complexity. Our Modified Main Algorithm is in essence the same as Chubanov's Main Algorithm, except that it uses our Modified Basic Procedure as a subroutine. It is shown that it has $O(n^4L)$ time complexity, just as in [1]. Some promising computational results are presented, in comparison with the optimization package Gurobi.

Keywords: linear homogeneous systems; algorithm; polynomial-time

1. Introduction

We deal with the problem

$$\begin{aligned} \text{find} \quad & x \in \mathbf{R}^n \\ \text{subject to} \quad & Ax = 0,\ x > 0, \end{aligned} \tag{1}$$

where A is an integer (or rational) matrix of size $m \times n$ and rank $(A) = m$.

Recently Chubanov [1] proposed a polynomial-time algorithm for solving this problem. A key ingredient for his algorithm is the so-called Basic Procedure (BP). As a result of the Basic Procedure we get either

(i) a feasible solution of (1), or
(ii) a feasible solution for the dual problem of (1), or
(iii) a cut for the feasible region of (1).

The cut in (iii) has the form $x_k \leq \frac{1}{2}$ for some index k and is used to rescale the matrix A. The rescaling happens in the Main Algorithm, which sends the rescaled matrix to the Basic Procedure until the Basic Procedure returns (i) or (ii).

*Email: c.roos@tudelft.nl

The dual problem in (ii) is

$$\begin{aligned} &\text{find} \quad u \in \mathbf{R}^m \\ &\text{subject to} \quad A^{\mathrm{T}}u \geq 0,\ A^{\mathrm{T}}u \neq 0. \end{aligned} \tag{2}$$

According to a variant of Farkas' lemma, due to Stiemke [15], the systems (1) and (2) form an alternative pair in the sense that exactly one of them is feasible. So if the Basic Procedure yields (ii) then (1) is infeasible. Since A has integer (or rational) entries, the number of calls of the Basic Procedure is polynomially bounded by $O(nL)$, where L denotes the bit size of the A. This follows from a classical result of Khachiyan [6] that gives a positive lower bound on the positive entries of a solution of a linear system of equations. The Basic Procedure needs at most $4n^3$ iterations per call and $O(n)$ time per iteration. So the overall time complexity becomes $O(n^5L)$. By performing a more careful analysis Chubanov reduced this bound by a factor n to $O(n^4L)$ [1, Theorem 2.1].

In this paper we present some improvements of Chubanov's method and its analysis. In [13, Section 1.2.2] we introduced a new way to derive cuts. We also proved that the new cuts are at least as sharp as the cuts used by Chubanov. In this paper we present in Section 2 (Lemma 2.2) a much simpler, almost obvious way to derive the same cuts. Section 3 is included not only to convince the reader that the new cuts are indeed sharper than the cuts in [1], but also because we need a biproduct (i.e. (11)) in Section 6.

The main contribution of this paper is the analysis in Section 4 of the Modified Basic Procedure (abbr. MBP) that uses the new cuts. It takes much more work than for Chubanov's cut to show that the MBP requires no more than $O(n^3)$ iterations. Though it improves Chubanov's bound with the factor 5, it does not improve the order.

A second improvement is in the search direction of the MBP. This search direction was also proposed in [13]; to make the paper self-supporting we include it in Section 4 (cf. Lemma 4.1). We were not able to improve the iteration bound of the MBP but in practice it makes much difference, as was acknowledged by Chubanov [1, Section 4.2].

In Section 5 we present our Modified Main Algorithm (abbr. MMA). In essence it is the same as Chubanov's Main Algorithm, except that it uses the Modified Basic Procedure as a subroutine. Its analysis is presented in Section 6. It is shown that the MMA solves problem (1) in $O(n^4L)$ time, just as in [1]. In Section 7 we present some computational results. We conclude with some comments in Section 8.

2. Preliminaries

Let $\mathcal{N}_A$ denote the null space of the $m \times n$ matrix A and $\mathcal{R}_A$ its row space. So

$$\mathcal{N}_A := \{x \in \mathbf{R}^n \,:\, Ax = 0\}, \quad \mathcal{R}_A := \{A^{\mathrm{T}}u \,:\, u \in \mathbf{R}^m\}.$$

We denote the orthogonal projections of $\mathbf{R}^n$ onto $\mathcal{N}_A$ and $\mathcal{R}_A$ as P_A and Q_A, respectively:

$$P_A := I - A^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1}A, \quad Q_A := A^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1}A.$$

Our assumption rank $(A) = m$ implies that the inverse of AA^{T} exists. Obviously we have

$$I = P_A + Q_A, \quad P_AQ_A = 0, \quad AP_A = 0, \quad AQ_A = A.$$

Now let $y \in \mathbf{R}^n$. In the sequel we use the notation

$$z = P_Ay, \quad v = Q_Ay.$$

So z and v are the orthogonal components of y in the spaces $\mathcal{N}_A$ and $\mathcal{R}_A$, respectively:

$$y = z + v, \quad z \in \mathcal{N}_A, \quad v \in \mathcal{R}_A.$$

These vectors play a crucial role in our approach. This is due to the following lemma.

LEMMA 2.1 *If $z > 0$ then z solves the primal problem* (1) *and if $0 \neq v \geq 0$ then v gives rise to a solution of the dual problem* (2).

Proof Since $z = P_A y$ we have $Az = AP_A y = 0$, because $AP_A = 0$. Hence the first statement in the lemma follows. The second statement follows by noting that $v \in \mathcal{R}_A$ implies $v = A^{\mathrm{T}}u$ for some u. Since A has full row rank, u is uniquely determined by v and solves (2). ■

Chubanov's approach heavily depends on the following observation [1]. If x is feasible for (1), then also $x' = x/\max(x)$ is feasible for (1), and this solution belongs to the unit cube, that is, $x' \in [0, 1]^n$. It follows that (1) is feasible if and only if the system

$$Ax = 0, \quad x \in (0, 1]^n \tag{3}$$

is feasible. Moreover, if $d > 0$ is a vector such that $x \leq d$ holds for every feasible solution of (3) then $x'' = x/d \leq e$, where x/d denote the entry-wise quotient of x and d, so $x''_i = x_i/d_i$ for each i. This means that x'' is feasible for the system

$$ADx = 0, \quad x \in (0, 1]^n, \tag{4}$$

where $D = \mathrm{diag}\,(d)$. Obviously, problem (4) is of the same type as problem (3), since it arises from (3) by rescaling A to AD. The algorithm presented below starts with $d = e$, and successively improves d by dividing one of its coordinates by 2. Like Chubanov's algorithm our algorithm can be seen as a systematic way to construct a sequence of vectors d such that $x \leq d$ holds for all feasible solutions of (3). However, while Chubanov used the vector z to construct cuts for (3), in this paper this is done by exploring properties of the vector v.

Before we sketch how this goes we introduce some notations. The vector that arises from v by replacing all its negative entries by zero is denoted as v^+. The vector v^- is defined in a similar way, so that $v^- = -(-v)^+$. Denoting the all-one vector of length n as $\mathbf{1}$, the sum of the positive entries in v is given by $\mathbf{1}^{\mathrm{T}}v^+$, and the sum of its negative entries by $\mathbf{1}^{\mathrm{T}}v^-$. We call y a *weak cutting vector* if

$$\mathbf{1}^{\mathrm{T}}v^+ < -\min(v) \quad \text{or} \quad -\mathbf{1}^{\mathrm{T}}v^- < \max(v). \tag{5}$$

The reason for this name is that if (5) holds then there exists at least one index k such that $x_k < 1$ holds for all solutions of (3). This is a consequence of the next lemma.

LEMMA 2.2 *Let x be feasible for* (3) *and $v = Q_A y$ for some y. Then every non-zero element v_k of v gives rise to a an upper bound for x_k, according to*

$$x_k \leq \mathbf{1}^{\mathrm{T}} \left[\frac{v}{-v_k} \right]^+. \tag{6}$$

Proof Since $Ax=0$ we have $P_A x = x$. Hence $v^{\mathrm{T}}x = v^{\mathrm{T}}P_A x = x^{\mathrm{T}}P_A v = x^{\mathrm{T}}P_A Q_A y = 0$, because $P_A Q_A = 0$. Now suppose $v_k < 0$. Then we deduce from $v^{\mathrm{T}}x = 0$ and $0 \le x \le \mathbf{1}$ that

$$-v_k x_k = \sum_{i \neq k} v_i x_i \le \sum_{i,\, v_i > 0} v_i x_i \le \sum_{i,\, v_i > 0} v_i = \mathbf{1}^{\mathrm{T}} v^{+}.$$

On the other hand, if $v_k > 0$ we obtain in the same way

$$v_k x_k = -\sum_{i \neq k} v_i x_i \le \sum_{i,\, v_i < 0} -v_i x_i \le \sum_{i,\, v_i < 0} -v_i = -\mathbf{1}^{\mathrm{T}} v^{-}.$$

Hence we have

$$x_k \le \begin{cases} \dfrac{\mathbf{1}^{\mathrm{T}} v^{+}}{-v_k} & \text{if } v_k < 0, \\ \dfrac{\mathbf{1}^{\mathrm{T}} v^{-}}{-v_k} & \text{if } v_k > 0. \end{cases} \tag{7}$$

These two results imply the inequality in the lemma, as one easily verifies. ■

COROLLARY 2.3 *If a non-zero entry v_k of v gives rise to a cut of the form $x_k \le \tau < 1$ then v_k has the same sign as $\mathbf{1}^{\mathrm{T}}v$.*

Proof Suppose $v_k < 0$ and $\mathbf{1}^{\mathrm{T}}v^{+}/(-v_k) = \tau < 1$ for some $\tau \ge 0$. This implies $\mathbf{1}^{\mathrm{T}}v^{+} + \tau v_k = 0$. Hence we may write

$$\mathbf{1}^{\mathrm{T}}v = \mathbf{1}^{\mathrm{T}}v^{+} + \mathbf{1}^{\mathrm{T}}v^{-} \le \mathbf{1}^{\mathrm{T}}v^{+} + v_k = \mathbf{1}^{\mathrm{T}}v^{+} + \tau v_k + (1-\tau)v_k = (1-\tau)v_k < 0.$$

It is left to the reader to verify (in the same way) that $v_k > 0$ and $\mathbf{1}^{\mathrm{T}}v^{-}/v_k = \tau < 1$ imply $\mathbf{1}^{\mathrm{T}}v > 0$. ■

In the sequel we shall use only cuts of the form $x_k \le \frac{1}{2}$. If the right-hand side expression in (6) does not exceed $\frac{1}{2}$ we call (6) a proper cut. Moreover, we call y a *proper cutting* vector if it induces at least one proper cut, otherwise we say that y is *weak cutting* if the smallest upper bound is less than 1 and else *non-cutting*. In the sequel we usually omit the word *proper*, so when we say that y is a cutting vector we always mean that it is a proper cutting vector.

An important observation is the following: the right-hand side in (6) does not change if we replace v by $-v$. More generally, it is homogeneous in v, because if we replace v by λv, where λ is any non-zero number, we get the same cut.

Example 2.4 By way of example we consider the case where v is given by

$$v = \begin{bmatrix} 3 \\ 4 \\ -2 \\ 0 \\ 2 \\ 6. \end{bmatrix}$$

Since $\mathbf{1}^{\mathrm{T}}v = 13 > 0$ only positive entries v_k in v may give rise to a non-void cut, by Corollary 2.3, and this happens if v_k exceeds

$$-\mathbf{1}^{\mathrm{T}}v^{-} = 2.$$

Thus we obtain a weak cut for x_1 and proper cuts for x_2 and x_6, namely:

$$x_1 \le \tfrac{2}{3}, \quad x_2 \le \tfrac{2}{4}, \quad x_6 \le \tfrac{2}{6}.$$

3. More on cut-generating vectors

In Lemma 2.2 we showed how to obtain a cut for problem (3) from a vector y. In this section we discuss two other methods to generate cuts from a given vector y and their relations to the cut defined in Lemma 2.2.

Fixing k, Chubanov [1, p. 692] considered the LO-problem

$$\max\{x_k \,:\, Ax = 0,\, x \in [0,1]^n\}.$$

The dual problem is

$$\min\{\mathbf{1}^{\mathrm{T}} w \,:\, A^{\mathrm{T}}\xi + w \geq e_k,\, w \geq 0\} = \min\{\mathbf{1}^{\mathrm{T}}\,[e_k - u]^+ \,:\, P_A u = 0\}.$$

The above equality uses that $u = A^{\mathrm{T}}\xi$ for some ξ if and only if $P_A u = 0$. Hence, if $y_k \neq 0$ we may take $u = v/y_k$, with v as defined Section 2. It then immediately follows from the Duality Theorem for Linear Optimization that

$$x_k \leq \mathbf{1}^{\mathrm{T}} \left[e_k - \frac{v}{y_k}\right]^+. \tag{8}$$

If moreover $y \geq 0$ it follows that

$$\left[e_k - \frac{v}{y_k}\right]^+ = \left[e_k - \frac{y - z}{y_k}\right]^+ \leq \left[\frac{z}{y_k}\right]^+,$$

because then $e_k - y/y_k \leq 0$. Hence we obtain

$$x_k \leq \mathbf{1}^{\mathrm{T}} \left[\frac{z}{y_k}\right]^+ = \frac{\mathbf{1}^{\mathrm{T}} z^+}{y_k} \leq \frac{\sqrt{n}\|z^+\|}{y_k} \leq \frac{\sqrt{n}\,\|z\|}{y_k}, \tag{9}$$

which is exactly the cut used in [1], where y is always non-negative.

We present yet another way to obtain the cuts in Lemma 2.2, thereby showing that these cuts are tighter than the cuts used by Chubanov. Instead of $u = v/y_k$ we use more generally $u = \alpha v$, with $\alpha \in \mathbf{R}$. We then have $x_k \leq q(\alpha)$ for every α, where the function $q(\alpha)$ is defined by

$$q(\alpha) := \mathbf{1}^{\mathrm{T}}\,[e_k - \alpha v]^+ = [1 - \alpha v_k]^+ + \sum_{i \neq k} [-\alpha v_i]^+, \quad \alpha \in \mathbf{R}.$$

One may easily verify that $q(\alpha)$ is a non-negative piecewise linear convex function with a breakpoint at $\alpha = 0$ and, if $v_k \neq 0$, another breakpoint at $\alpha = 1/v_k$. Since $q(\alpha)$ is convex it attains its minimal value at a breakpoint. The breakpoint at $\alpha = 0$ yields the void inequality $x_k \leq q(0) = 1$. So only the breakpoint at $\alpha = 1/v_k$ is of interest, and this yields exactly the inequality in Lemma 2.2 (because the first term in the expression for $q(\alpha)$ vanishes at this breakpoint).

We conclude from the above analysis that for each non-zero y and for each k one has

$$x_k \leq \min\left(1, \sum_{i=1}^{n} \left[\frac{-v}{v_k}\right]^+\right) \leq \mathbf{1}^{\mathrm{T}} \left[e_k - \frac{v}{y_k}\right]^+ \leq \frac{\sqrt{n}\,\|z\|}{y_k}, \tag{10}$$

where the first inequality assumes $v_k \neq 0$, the second inequality $y_k \neq 0$ and the third inequality $y \geq 0$ and $y_k > 0$.

Of course, an upper bound is non-void if and only if its value is less than 1. Note that the second inequality yields a non-void cut only if v_k has the same sign as y_k. This easily follows because otherwise the value of the kth term alone in this expression already is at least 1.

We finally mention that (10) implies a result that we need later on, namely

$$\sum_{i=1}^{n} \left[\frac{-v}{v_k}\right]^+ > \frac{1}{2} \;\Rightarrow\; y_k < 2\sqrt{n}\,\|z\|\,, \tag{11}$$

provided that $y \geq 0$.

4. Modified basic procedure

We start by reformulating the dual problem (2) in terms of the vector $y = A^{\mathrm{T}}u$. Since $y = A^{\mathrm{T}}u$ holds for some u if and only if $P_A y = 0$, it follows that the dual problem is feasible if and only if the system

$$P_A y = 0, \quad y \geq 0, \quad y \neq 0 \tag{12}$$

has a solution. Chubanov's algorithm can be viewed as a systematic search method for a vector y satisfying (12). It will be convenient to call any such vector a *dual feasible vector*.

Since (12) is homogeneous in y and $y \neq 0$, we may restrict the search to vectors y such that $\mathbf{1}^{\mathrm{T}}y = 1$, where $\mathbf{1}$ denotes the all-one vector. If during this search it happens that $P_A y > 0$, then $z = P_A y$ is a solution of (1). This follows because $AP_A = 0$, whence $Az = 0$. If this happens we call the vector y *primal feasible*.

From now on y always denotes a positive vector such that $\mathbf{1}^{\mathrm{T}}y = 1$. In this section we show that if y is not primal or dual feasible then it is possible to find in $O(n^3)$ time a new vector y such that one of the following three cases occurs:

(i) $z = P_A y$ is feasible for (1);
(ii) $z = 0$, meaning that y satisfies (12);
(iii) y is a (proper) cutting vector.

In the first two cases the status of (1) is clear: in case (i) we have a solution of (3), and in case (ii) a certificate for its infeasibility. In case (iii) y induces for at least one index k an inequality $x_k \leq \frac{1}{2}$ for all solutions of (3). Obvious such an inequality cuts off halve of the feasible region of (3). It enables us to update the current vector d by dividing its kth entry by 2.

Our algorithm is presented in Algorithm 1; it is a modified version of Chubanov's Basic Procedure [1]. We call it Modified Basic Procedure and refer to it with the abbreviation MBP.

The MBP uses as input the projection matrix P_A and the vector y, with $y > 0$ and $\mathbf{1}^{\mathrm{T}}y = 1$. The notation bound$_j(y)$ stands for the upper bound for x_j in Lemma 2.2. So

$$\text{bound}_j(y) = \begin{cases} \dfrac{-\mathbf{1}^{\mathrm{T}}v^-}{v_j} & \text{if } v_j > 0, \\[2ex] \dfrac{\mathbf{1}^{\mathrm{T}}v^+}{-v_j} & \text{if } v_j < 0. \end{cases}$$

The smallest of these bounds is denoted as bound (y). More precisely,

$$\text{bound}\,(y) = \begin{cases} \dfrac{-\mathbf{1}^{\mathrm{T}}v^-}{\max(v^+)} & \text{if } \mathbf{1}^{\mathrm{T}}v > 0, \\[2ex] \dfrac{\mathbf{1}^{\mathrm{T}}v^+}{-\min(v^-)} & \text{if } \mathbf{1}^{\mathrm{T}}v < 0. \end{cases}$$

Note that each of these quantities can be computed in $O(n)$ time.

If the vector y is primal feasible or dual feasible and bound $(y) > \frac{1}{2}$ the MBP requires only one iteration. Then the output is y (unchanged), $\bar{y} = 0$, $z = P_A y$ and case $= 1$, or 2, respectively. Otherwise it generates a new vector y such that one of the three cases (i), (ii) or (iii) occurs as we now will show.

If y is such that the status of (1) is not yet decided (i.e. case $= 0$) then $z \neq 0$ and at least one component of z is negative or zero. Hence we may find a non-empty set K of indices such that

$$\sum_{k \in K} z_k \leq 0.$$

Denoting the kth column of P_A as p^k, we have $p^k = P_A e_k$, where e_k denotes the kth unit vector. We define

$$e_K := \frac{1}{|K|} \sum_{k \in K} e_k, \quad p_K := P_A e_K = \frac{1}{|K|} \sum_{k \in K} p^k. \tag{13}$$

Note that $0 \neq e_K \geq 0$, and $\mathbf{1}^{\mathrm{T}} e_K = 1$. If $p_K = 0$ ($p_K > 0$), then e_K is dual (primal) feasible and we are done. Hence, we may assume that $p_K \neq 0$. Using again that P_A is a projection matrix we obtain $P_A z = P_A^2 y = P_A y = z$. This implies $z^{\mathrm{T}} p^k = z^{\mathrm{T}} P_A e_k = z^{\mathrm{T}} e_k = z_k$ for each k. Thus we obtain

$$z^{\mathrm{T}} p_K = \frac{1}{|K|} \sum_{k \in K} z^{\mathrm{T}} p^k = \frac{1}{|K|} \sum_{k \in K} z_k \leq 0.$$

As a consequence, in the equation

$$\|z - p_K\|^2 = \left(\|z\|^2 - z^{\mathrm{T}} p_K\right) + \left(\|p_K\|^2 - z^{\mathrm{T}} p_K\right) \tag{14}$$

the two bracketed terms are both positive, because z and p_K are non-zero and $z^{\mathrm{T}} p_K \leq 0$. Therefore, we may define a new y-vector, denoted by $\tilde{y}$, according to

$$\tilde{y} = \alpha y + (1 - \alpha) e_K, \quad \alpha = \frac{\|p_K\|^2 - z^{\mathrm{T}} p_K}{\|z - p_K\|^2} = \frac{p_K^{\mathrm{T}}(p_K - z)}{\|z - p_K\|^2}. \tag{15}$$

Because of (14), α is well-defined and $\alpha \in (0, 1)$. Since $y > 0$ and $e_K \geq 0$, we may conclude that $\tilde{y} > 0$ and, since $\mathbf{1}^{\mathrm{T}} y = \mathbf{1}^{\mathrm{T}} e_K = 1$, also $\mathbf{1}^{\mathrm{T}} \tilde{y} = 1$.

The transformation (15) from y to $\tilde{y}$ is the key element in Algorithm 1.

It iterates (15) until y is primal feasible or dual feasible or a cutting vector. Our next step is to find an upper bound for the number of iterations of the MBP. For this the next two lemmas are important. The first lemma measures progress in terms of the merit function $1/\|\tilde{z}\|^2$.

LEMMA 4.1 *Let $z \neq 0$ and let K be such that $\sum_{k \in K} z_k \leq 0$ and $p_K \neq 0$. With $\tilde{y}$ as in* (15) *and $\tilde{z} := P_A \tilde{y}$, one has*

$$\frac{1}{\|\tilde{z}\|^2} \geq \frac{1}{\|z\|^2} + |K|\,. \tag{16}$$

Algorithm 1: $[y, \bar{y}, z, J, \text{case}] = \text{MODIFIED BASIC PROCEDURE}(P_A, y)$

INITIALIZE: $z = P_A y$; $\bar{y} = 0$; case = 0; $J = \emptyset$;
while bound$(y) > \frac{1}{2}$ ***and*** case = 0 **do**
  if $z > 0$ **then**
    case $= 1$ (y is primal feasible); **return**
  else
    if $z = 0$ **then**
      case $= 2$ (y is dual feasible); **return**
    else
      find $K \neq \emptyset$ such that $\sum_{k \in K} z_k \leq 0$
      if $p_K > 0$ **then**
        $y = e_K$
        case $= 1$ (e_K is primal feasible); **return**
      else
        if $p_K = 0$ **then**
          $y = e_K$
          case $= 2$ (e_K is dual feasible); **return**
        else
          $\bar{y} := y$
          $\alpha = p_K^{\mathrm{T}}(p_K - z)/\|z - p_K\|^2$
          $y = \alpha y + (1 - \alpha)e_K$
          $z = \alpha z + (1 - \alpha)p_K$ $(= P_A y)$
        end
      end
    end
  end
end
if case $= 0$ **then**
  find a non-empty set J such that $J \subseteq \{j : \text{bound}_j(y) \leq \frac{1}{2}\}$
end

Proof We have

$$\tilde{z} = \alpha P_A y + (1 - \alpha) P_A e_K = \alpha z + (1 - \alpha) p_K = p_K + \alpha(z - p_K).$$

Hence,

$$\|\tilde{z}\|^2 = \alpha^2 \|z - p_K\|^2 + 2\alpha {p_K}^{\mathrm{T}}(z - p_K) + \|p_K\|^2.$$

The value of α that minimizes this expression is given in (15). It follows that

$$\|\tilde{z}\|^2 = \|p_K\|^2 - \frac{\left[p_K^{\mathrm{T}}(z - p_K)\right]^2}{\|z - p_K\|^2} = \frac{\|p_K\|^2\|z\|^2 - (z^{\mathrm{T}}p_K)^2}{\|p_K\|^2 + \|z\|^2 - 2z^{\mathrm{T}}p_K} \leq \frac{\|p_K\|^2\|z\|^2}{\|z\|^2 + \|p_K\|^2},$$

where we used $z^{\mathrm{T}}p_K \leq 0$. Since P_A is a projection matrix, $\|P_A e_K\| \leq \|e_K\|$. So we may write

$$\|p_K\|^2 = \|P_A e_K\|^2 \leq \|e_K\|^2 = \left\| \frac{1}{|K|} \sum_{k \in K} e_k \right\|^2 = \frac{1}{|K|^2} \left\| \sum_{k \in K} e_k \right\|^2 = \frac{|K|}{|K|^2} = \frac{1}{|K|}. \qquad (17)$$

It follows that

$$\frac{1}{\|\tilde{z}\|^2} \geq \frac{1}{\|z\|^2} + \frac{1}{\|p_K\|^2} \geq \frac{1}{\|z\|^2} + |K|\,, \tag{18}$$

as desired. ■

Below we derive an upper bound for $1/\|z\|^2$ if y is not a cutting vector. Thus we assume that bound$(y) \geq 1/\sigma$ for some $\sigma \geq 1$. Then we have bound$_k(y) \geq 1/\sigma$ for all k such that $v_k \neq 0$. This means that

$$\mathbf{1}^{\mathrm{T}} \left[\frac{v}{-v_k}\right]^+ \geq \frac{1}{\sigma}, \quad \forall k \text{ such that } v_k \neq 0. \tag{19}$$

If $v \geq 0$ or $v \leq 0$ then the left-hand side expression in (19) equals zero. Hence, (19) implies that v must have both positive and negative entries. The set of all (non-zero) vectors in $\mathbf{R}^n$ that satisfy (19) is denoted as V_σ. So we have

$$V_\sigma = \{v \in \mathbf{R}^n \setminus \{0\} \,:\, v \text{ satisfies (19)}\}.$$

This definition implies that y is not a (proper) cutting vector if and only if the vector v belongs to V_2. As we made clear before, we may assume without loss of generality that $\mathbf{1}^{\mathrm{T}} v \geq 0$. Then only positive entries in v can give rise to a cut. Therefore $v \in V_\sigma$ holds if and only if

$$-\sigma \mathbf{1}^{\mathrm{T}} v^- \geq \max(v). \tag{20}$$

The definitions of y, v and z imply the following relations:

$$y \geq 0,\ \mathbf{1}^{\mathrm{T}} y = 1,\ y = z + v,\ z^{\mathrm{T}} v = 0, \tag{21}$$

where $v \in V_\sigma$. Our aim is to derive a positive lower bound for $\|z\|$ if $v \in V_\sigma$. Fixing $v \in V_\sigma$, we therefore consider the minimization problem

$$\min_{y,z,\beta} \{\|z\| \,:\, y \geq 0,\ \mathbf{1}^{\mathrm{T}} y = 1,\ y = z + \beta v,\ z^{\mathrm{T}} v = 0\}. \tag{22}$$

We introduced an additional variable β because if $\beta = 1$ problem (22) may be infeasible.[1] A crucial observation is that if $\beta \neq 0$ then $v \in V_\sigma$ if and only if $\beta v \in V_\sigma$. Another important fact is that the problem is easy to solve if $\beta = 0$, because then $z = y$. Since $y \geq 0$ and $\mathbf{1}^{\mathrm{T}} y = 1$ we then have $\|z\| \geq 1/\sqrt{n}$, whence $1/\|z\|^2 \leq n$. The main result in this section is the following lemma, whose proof makes clear that much smaller values of $\|z\|$ are achieved if $\beta \neq 0$.

Lemma 4.2 *Let $n \geq 2$, $\sigma \geq 2$ and $v \in V_\sigma$. If y and z satisfy* (21) *then*

$$\frac{1}{\|z\|^2} \leq \frac{n^3 \sigma^2}{5}.$$

Proof This proof uses a second optimization problem, namely [2]

$$\max_{\alpha,\lambda}\{\alpha \ : \ \lambda \geq \alpha \mathbf{1},\ \|\lambda\| \leq 1,\ \lambda^{\mathrm{T}} v = 0\}. \tag{23}$$

The relevance of this problem for our purpose is that if (y, z, β) is feasible for (22) and (λ, α) for (23), then one has

$$\|z\| \geq \|z\|\,\|\lambda\| \geq \lambda^{\mathrm{T}} z = \lambda^{\mathrm{T}}(y - \beta v) = \lambda^{\mathrm{T}} y \geq \alpha \mathbf{1}^{\mathrm{T}} y = \alpha, \tag{24}$$

where we used the Cauchy–Schwarz inequality and the feasibility conditions for both problems. Hence, if we have a feasible solution (λ_v, α_v) for each $v \in V_\sigma$, then it follows that

$$\|z\| \geq \min\{\alpha_v \ : \ v \in V_\sigma\}. \tag{25}$$

This argument underlies the rest of the proof and leads to the upper bound for $1/\|z\|^2$ in the lemma.

Let $v \in V_\sigma$. It will be convenient to introduce the index sets R and S as follows:

$$R = \{i \ : \ v_i > 0\}, \quad S = \{i \ : \ v_i < 0\}.$$

We define the vector $\lambda \in \mathbf{R}^n$ as follows:

$$\lambda_i = \begin{cases} \tau & \text{if } i \in R \\ \xi & \text{if } i \in S \\ \tau & \text{if } v_i = 0. \end{cases} \tag{26}$$

Denoting the restriction of λ to the index set S as λ_S and using similar notation for the restrictions of λ, v and $\mathbf{1}$ to a set of indices, we may write $\lambda_R = \tau \mathbf{1}_R$ and $\lambda_S = \xi \mathbf{1}_S$. Now λ is feasible for (23) if for some α:

$$\xi \mathbf{1}_S{}^{\mathrm{T}} v_S + \tau \mathbf{1}_R{}^{\mathrm{T}} v_R = 0, \tag{27}$$

$$|S|\,\xi^2 + |R|\,\tau^2 = 1. \tag{28}$$

$$\lambda \geq \alpha \mathbf{1}. \tag{29}$$

The definitions of the sets S and R imply $\mathbf{1}_R^{\mathrm{T}} v_R > 0$ and $\mathbf{1}_S^{\mathrm{T}} v_S < 0$. Since $0 \leq \mathbf{1}^{\mathrm{T}} v = \mathbf{1}_S^{\mathrm{T}} v_S + \mathbf{1}_R^{\mathrm{T}} v_R$ we have $\mathbf{1}_R^{\mathrm{T}} v_R \geq -\mathbf{1}_S^{\mathrm{T}} v_S$. Since (27) holds if and only if

$$\xi = \frac{\mathbf{1}_R^{\mathrm{T}} v_R}{-\mathbf{1}_S^{\mathrm{T}} v_S} \tau. \tag{30}$$

we must have $\xi \geq \tau$. Then (29) holds if $\alpha = \tau$. Hence λ is feasible for (23) with objective value τ if (28) holds. This is true if and only if

$$|S|\,\frac{(\mathbf{1}_R^{\mathrm{T}} v_R)^2}{(\mathbf{1}_S^{\mathrm{T}} v_S)^2} \tau^2 + |R|\,\tau^2 = 1,$$

which is equivalent to

$$\tau^2 = \frac{(\mathbf{1}_S^{\mathrm{T}} v_S)^2}{|R|\,(\mathbf{1}_S^{\mathrm{T}} v_S)^2 + |S|\,(\mathbf{1}_R^{\mathrm{T}} v_R)^2}. \tag{31}$$

At this stage we use $v \in V_\sigma$, that is, (20). Since $\mathbf{1}^{\mathrm{T}} v^- = \mathbf{1}_S^{\mathrm{T}} v_S$ we derive from (20) that each v_i with $i \in R$ satisfies $-\sigma \mathbf{1}_S^{\mathrm{T}} v_S \geq v_i > 0$. So we may write

$$0 < \mathbf{1}_R{}^{\mathrm{T}} v_R = \sum_{i \in R} v_i \leq \sum_{i \in R} -\sigma \mathbf{1}_S{}^{\mathrm{T}} v_S = -\sigma \mathbf{1}_S{}^{\mathrm{T}} v_S \sum_{i \in R} 1 = -\sigma\, |R|\, \mathbf{1}_S{}^{\mathrm{T}} v_S.$$

Substitution into (31) yields

$$\tau^2 \geq \frac{(\mathbf{1}_S^{\mathrm{T}} v_S)^2}{|R|\,(\mathbf{1}_S^{\mathrm{T}} v_S)^2 + \sigma^2\,|S|\,|R|^2\,(\mathbf{1}_S^{\mathrm{T}} v_S)^2} = \frac{1}{|R|\left(1 + \sigma^2\,|S|\,|R|^2\right)}. \tag{32}$$

Hence, by (25),

$$\frac{1}{\|z\|^2} \leq \max_{R,S} |R| \left(1 + \sigma^2\,|S|\,|R|^2\right).$$

It remains to find out how large the last expression can be. This is possible because of the obvious inequality $|R| + |S| \leq n$. Since the expression is decreasing in both $|R|$ and $|S|$ the largest value occurs if $|R| + |S| = n$. Therefore, putting $t = |R|$ and $|S| = n - t$ we need to find the maximal value of the function

$$f(t) = t(1 + \sigma^2 t(n - t)), \quad 1 \leq t \leq n - 1.$$

One easily verifies that the largest value of $f(t)$ occurs if $t = \theta$, with $\theta = (m + \sqrt{3 + m^2})/3\sigma$, where $m = n\sigma$, and then the value is given by

$$\begin{aligned} f(\theta) &= \frac{\left(m + \sqrt{3 + m^2}\right)\left(6 + m^2 + m\sqrt{3 + m^2}\right)}{27\sigma} \\ &= \frac{2m^3 + 9m + \left(6 + 2m^2\right)\sqrt{3 + m^2}}{27\sigma}. \end{aligned}$$

Since $n \geq 2$ and $\sigma \geq 2$ we have $m \geq 4$, whence $\sqrt{3 + m^2} \leq m + \frac{9}{25}$. Hence we get

$$f(\theta) \leq \frac{2m^3 + 9m + \left(6 + 2m^2\right)\left(m + \frac{9}{25}\right)}{27\sigma} = \frac{4m^3 + \frac{18}{25}m^2 + 15m + \frac{54}{25}}{27\sigma}.$$

Since $m \geq 4$, we have $\frac{18}{25}m^2 + 15m + \frac{54}{25} \leq 1.4\, m^3$. Substitution gives

$$f(\theta) \leq \frac{5.4\, m^3}{27\sigma} = \frac{m^3}{5\,\sigma} = \frac{n^3\sigma^3}{5\,\sigma} = \frac{n^3\sigma^2}{5}.$$

This implies the inequality in the lemma. ■

Our interest is the case where $\sigma = 2$. Then Lemma 4.2 yields that if y is non-cutting then

$$\frac{1}{\|z\|^2} \leq \frac{4n^3}{5} < n^3. \tag{33}$$

It may be worth noting that this improves the upper bound for $1/\|z\|^2$ in [1, Lemma 2.2] by a factor 5.

THEOREM 4.3 *After at most n^3 iterations the MBP yields a vector y that is either a cutting vector (case = 0) or primal feasible (case = 1) or dual feasible (case = 2).*

Proof If bound $(y) \leq \frac{1}{2}$ then y is a cutting vector and the MBP requires only 1 iteration. Otherwise bound $(y) > \frac{1}{2}$, which implies $1/\|z\|^2 < n^3$, by (33). If during the execution of the while loop in Algorithm 1 it happens that $z > 0$ or $z = 0$ then the MBP immediately stops. Otherwise, since $|K| \geq 1$, the while loop increases $1/\|z\|^2$ by at least 1, by Lemma 4.1. Hence, after at most n^3 executions of the while loop the algorithm yields a vector y that is primal feasible (case $= 1$) or dual feasible (case $= 2$) or such that $1/\|z\|^2 \geq n^3$. In the last case it follows from Lemma 4.2 that y is a cutting vector (case $= 0$). ■

Provided that we take care that $|K| = O(1)$, each execution of the while loop requires at most $O(n)$ time. Therefore each execution of the MBP will require at most $O(n^4)$ time. Note that this bound is valid only if the size of the set J in line 23 of the MBP is also of order 1, because the computation of bound$_j(y)$ requires $O(n)$ time for each element of J. Therefore, we assume below always that the set J is chosen in a such a way that $|J| = O(1)$.

In order to solve (1) one needs to call the MBP several times by another algorithm, named the Modified Main Algorithm, a modified version of Chubanov's Main Algorithm [1]. We deal with this in the next section. Then it will become clear why the output of the MBP contains the vector $\bar{y}$. One easily verifies that $\bar{y}$ is the zero vector if the MBP requires only one iteration; otherwise it is the last non-cutting vector y generated during the course of the MBP.

5. Modified main algorithm

As announced in Section 2 the MMA maintains a vector vector d such that $x \leq d$ holds for every feasible solution of problem (3). Initially d is the all-one vector. But each time the MBP generates a cutting vector the upper bound d_j for x_j can be divided by 2, for all indices j in the set J.

As a consequence, the entries of d have the form 2^{-t_i}, where t_i denotes the number of times that a cut was generated for the i-th entry of x. Hence we may restate (4) in the following way:

$$Ax = 0, \quad 0 < x_i \leq d_i, \quad 1 \leq i \leq n, \tag{34}$$

where $d_i = 2^{-t_i}$. According to Khachiyan's result [6] there exists a positive number τ satisfying $1/\tau = O(2^L)$, where L denotes the bit size of the matrix A, such that the positive coordinates of the basic feasible solutions of (3) are bounded from below by τ [6,12,14].

Since the basic feasible solutions also satisfy $x \leq d$, we conclude that (3), and hence problem (1), must be infeasible as soon as $d_i < \tau$ for some i. This explains the statement in line 7 of Algorithm 2. As a consequence of this line the MMA will stop if problem (3) turns out to be infeasible due to Khachiyan's criterion (case = 3).

The MMA starts with $d = e$ and $y = e/n$. As long as the status of problem (1) is not yet fixed (i.e. case = 0) each execution of the while loop does the following. Given the current matrix A the projection matrix P_A is computed. Then the MBP is called. If the MBP yields case > 0 the algorithm stops. If case $= 1$, the vector z is positive and satisfies $ADz = 0$, whence $x = Dz$ solves problem (1) and if case = 2 the problem is infeasible (or more precisely, has no solution x satisfying $x \geq \tau \mathbf{1}$). Otherwise, if case = 0, it divides the entries of d indexed by the set J by 2 and then checks if one of the new entries in d is smaller than τ. If so, it stops (with case = 3). Otherwise we still have case = 0. So far everything goes as one might expect.

At this stage the auxiliary vector $\bar{y}$ enters the scene. Without this vector the algorithm would still work correctly, but with it the runtime can be guaranteed via Lemmas 6.1 and 6.2 in the next section. As mentioned before this vector equals the zero vector if the MBP did not change the

Algorithm 2: $[x, y, d, \text{case}]$ =MODIFIED MAIN ALGORITHM(A, τ)

INITIALIZE: $d = e$; $y = e/n$; $x = 0$; case $= 0$;
while *case* $= 0$ **do**
  $P_A = I - A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}A$
  $[y, \bar{y}, z, J, \text{case}] =$ Modified Basic Procedure(P_A, y)
  if case $= 0$ **then**
    $d_J = d_J/2$
    if $\min(d_J) < \tau$ **then**
      case $= 3$
    else
      if $\bar{y} \neq 0$ **then**
        $y = \bar{y}$
      end
      $A_J = A_J/2$
      $y_J = y_J/2$
      $y = y/\mathbf{1}^{\mathrm{T}}y$
    end
  end
end
if case $= 1$ **then**
  $D = \text{diag}(d)$
  $x = Dz$
end

vector y, but otherwise it is the last non-cutting vector generated by the MBP. The current y – which is a cutting vector with respect to the current A – is replaced by the non-cutting vector $\bar{y}$.

Next the MMA divides the columns of A and the entries of y indexed by the set J by 2. As a consequence the constraint matrix equals AD, with $D = \text{diag}\,(d)$ (where A is the original matrix and d the current vector of upper bounds for the entries of feasible vectors x). Finally the MMA normalizes y. After this the while loop is entered again. So the P_A is computed for the new matrix A, etc.

6. Complexity analysis

Due to the use of Khachiyan's result we can easily derive an upper bound for the number of iterations of the MMA. As we noticed in the previous section, during the course of the MMA we certainly have $t_i \leq \log_2 \frac{1}{\tau}$ for each i. Let T denote the number of times that the MMA calls the MBP. Then T is also equal to the number of returns from the MBP to the MMA. Since each return, except possibly the last one, yields at least one cut, we must have

$$T \leq 1 + \sum_{i=1}^{n} t_i.$$

Hence we get

$$T \leq 1 + \sum_{i=1}^{n} \log_2 \frac{1}{\tau} = 1 + n \log_2 \frac{1}{\tau} = O(nL). \tag{35}$$

Since the MBP needs at most n^3 iterations, by Theorem 4.3, in total we need $O(n^4L)$ MBP-iterations. Each MBP-iteration needs $O(n)$ time. Hence, the contribution of the MBP to the time complexity of the MMA becomes $O(n^5L)$.

The main computational task in the MMA is the computation of P_A. The first time this can be done in $O(n^3)$ time [3,12]. Since $|J| = O(1)$, in each next iteration the matrix A is a low-rank modification of the previous matrix A. By applying the Sherman–Morrisen–Woodbury formula [5]

$$(A + aa^{\mathrm{T}})^{-1} = A^{-1} - \left(1 + a^{\mathrm{T}}A^{-1}a\right)^{-1} A^{-1}aa^{\mathrm{T}}A^{-1}$$

$|J|$ times, the new projection matrix P_A can be computed in $O(n^2)$ time. So, in total the MMA needs

$$O(n^3) + O(n^2)O(nL) = O(n^3L) \tag{36}$$

time. This yields the overall time complexity $O(n^5L) + O(n^3L) = O(n^5L)$.

Clearly the time estimate for the MBP is worse than for the MMA. We conclude the paper by proving that the time complexity for the MBP can be improved by a factor n, thus yielding an overall time complexity of $O(n^4L)$.

Crucial for our result is the next lemma. It deals with the case where the MMA redefines y in line 10-14. There $\bar{y}$ is a non-cutting vector and y is obtained by rescaling $\bar{y}$ to $D\bar{y}$ and then normalizing y so that $\mathbf{1}^{\mathrm{T}}y = 1$, with $D = \mathrm{diag}\,(d)$ and $d_i = \frac{1}{2}$ if $i \in J$ and $d_i = 1$ if $i \notin J$. More generally we will assume that $d_i \leq \frac{1}{2}$ for $i \in J$. Moreover, A will denote the current version (maybe already rescaled) of the initial matrix A. The lemma slightly improves [1, Lemma 2.3]; it differs from that lemma only in the assumption $d_i \leq \frac{1}{2}$ for $i \in J$.

LEMMA 6.1 *Let $\bar{y}$ be non-cutting with respect to (the current matrix) A, D as just defined and $y = D\bar{y}/\mathbf{1}^{\mathrm{T}}D\bar{y}$. If $\bar{z} = P_A\bar{y}$ and $z = P_{AD}y$, then*

$$\frac{1}{\|\bar{z}\|^2} - \frac{1}{\|z\|^2} < 2\,|J|\,n^2.$$

Proof We start by proving the inequality $\|P_{AD}D\bar{y}\| \leq \|P_A\bar{y}\| = \|\bar{z}\|$. Since $\bar{v} := \bar{y} - \bar{z} \in \mathcal{R}_A$ we have $\bar{v} = A^{\mathrm{T}}u$ for some u. Using $P_{AD}DA^{\mathrm{T}} = 0$ it follows that $P_{AD}D\bar{v} = 0$. Hence $P_{AD}D(\bar{y} - \bar{z}) = 0$, whence $P_{AD}D\bar{y} = P_{AD}D\bar{z}$. Since P_{AD} is a projection matrix, it does not increase the length of a vector. Therefore, also using $0 \leq d \leq \mathbf{1}$ we obtain

$$\|P_{AD}D\bar{y}\| \leq \|D\bar{z}\| \leq \|\bar{z}\|.$$

Also using the definitions of y and z it follows that

$$\|z\| = \|P_{AD}y\| = \left\|P_{AD}\frac{D\bar{y}}{\mathbf{1}^{\mathrm{T}}D\bar{y}}\right\| = \frac{1}{\mathbf{1}^{\mathrm{T}}D\bar{y}}\,\|P_{AD}D\bar{y}\| \leq \frac{\|\bar{z}\|}{\mathbf{1}^{\mathrm{T}}D\bar{y}}.$$

Since $\mathbf{1}^{\mathrm{T}}\bar{y} = 1$, $d_i = 1$ if $i \notin J$ and $d_i \leq \frac{1}{2}$ if $i \in J$, we may write

$$0 \leq \mathbf{1}^{\mathrm{T}}D\bar{y} \leq \sum_{i\notin J}\bar{y}_i + \sum_{i\in J}\frac{1}{2}\bar{y}_i = 1 - \sum_{i\in J}\bar{y}_i + \sum_{i\in J}\frac{1}{2}\bar{y}_i = 1 - \sum_{i\in J}\frac{1}{2}\bar{y}_i.$$

We therefore obtain

$$\frac{1}{\|z\|^2} \geq \frac{\left(\mathbf{1}^{\mathrm{T}}D\bar{y}\right)^2}{\|\bar{z}\|^2} \geq \frac{\left(1 - \frac{1}{2}\sum_{i\in J}\bar{y}_i\right)^2}{\|\bar{z}\|^2} \geq \frac{1}{\|\bar{z}\|^2} - \frac{\sum_{i\in J}\bar{y}_i}{\|\bar{z}\|^2}.$$

Since $\bar{y}$ is non-cutting, we derive from (11) that

$$\bar{y}_k < 2\,\|\bar{z}\|\,\sqrt{n}, \quad 1 \le k \le n.$$

Using this and $1/\|\bar{z}\| \le n\sqrt{n}$, by (33), we obtain

$$\frac{1}{\|\bar{z}\|^2} - \frac{1}{\|z\|^2} \le \frac{\sum_{i\in J}\bar{y}_i}{\|\bar{z}\|^2} \le \frac{2\,|J|\,\|\bar{z}\|\,\sqrt{n}}{\|\bar{z}\|^2} = \frac{2\,|J|\,\sqrt{n}}{\|\bar{z}\|} \le 2\,|J|\,n^2.$$

This proves the lemma. ■

Lemma 6.1 makes it possible to improve the upper bound for the total number of MBP-iterations. Following Chubanov [1], we distinguish two types of MBP-iterations: *slow* iterations when the MBP changes $\bar{y}$ – and also y – and *fast* iterations that leave $\bar{y}$ and y unchanged. In short, a MBP-iteration is slow if and only if it yields $\bar{y} \neq 0$.

The number of fast iterations is denoted as N_f and the number of slow iterations as N_s. So the total number of MBP-iterations equals $N := N_f + N_s$.

Since case $= 0$ at the start of the MBP, a fast iteration occurs if and only if at the start of the while loop in the MBP y is a proper cutting vector or y (or e_K) is primal or dual feasible. Hence, it is either the last MBP-iteration or it generates cuts for the indices in J, without changing y. From this one easily understands that $N_f = T$, where T is the total number of MMA-iterations. Since $T = O(nL)$, by (35), we obtain

$$N = O(nL) + N_s.$$

In order to obtain an upper bound for N_s we number the MBP iterations (including the fast ierations) from 1 to N. We define iteration numbers $a_1, \ldots a_k$ and $b_1, \ldots b_k$ in such a way that $a_i \le b_i < a_{i+1}$ for $1 \le i < k$ and if $a_i \le j \le b_i$ for some i then iteration j is slow, and otherwise iteration j is fast. In other words, the sequence of MBP iterations contains k 'trains' of slow sequences $[a_i, b_i]$ that are separated by one or more fast MBP-iterations (see Figure 1). Note that fast iterations may also occur before the first train and after the last train.

It may happen that $k = 0$, that is, all MBP iterations are fast. Then $N = O(nL)$, and the MBP needs $O(n^2L)$ time in total. So we assume below that $k \ge 1$. We denote the y- and z-vector at the start of the while loop at iteration j as y_j and z_j, respectively. Then we have the following result.

Lemma 6.2 *For each i such that $1 \le i \le k$ one has*

$$b_i - a_i \le \frac{1}{\|z_{b_i}\|^2} - \frac{1}{\|z_{a_i}\|^2}.$$

Proof One has

$$b_i - a_i = \sum_{j=a_i}^{b_i-1} 1 \le \sum_{j=a_i}^{b_i-1} \left(\frac{1}{\|z_{j+1}\|^2} - \frac{1}{\|z_j\|^2} \right) = \frac{1}{\|z_{b_i}\|^2} - \frac{1}{\|z_{a_i}\|^2},$$

where the inequality is due to Lemma 4.1 and the equalities are obvious. ■

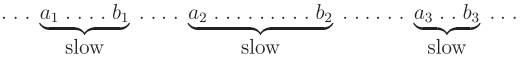

Figure 1. Sequence of BMP iterations.

It follows from Lemma 6.2 that the total number of slow MBP iterations satisfies

$$N_s \leq \sum_{i=1}^{k} \left(\frac{1}{\|z_{b_i}\|^2} - \frac{1}{\|z_{a_i}\|^2} \right).$$

By rearranging the terms in the above sum we obtain

$$N_s \leq \frac{1}{\|z_{b_k}\|^2} - \frac{1}{\|z_{a_1}\|^2} + \sum_{i=1}^{k-1} \left(\frac{1}{\|z_{b_i}\|^2} - \frac{1}{\|z_{a_{i+1}}\|^2} \right).$$

Since iteration b_k is slow, the y-vector at the start of the while loop is non-cutting. Due to (33) this implies that the first term in the last expression does not exceed n^3. Neglecting the second term we obtain

$$N_s \leq n^3 + \sum_{i=1}^{k-1} \left(\frac{1}{\|z_{b_i}\|^2} - \frac{1}{\|z_{a_{i+1}}\|^2} \right). \tag{37}$$

If $k=1$, this gives $N_s \leq n^3$, in accordance with Theorem 4.3. Then $N \leq O(nL)+n^3$, which implies that the MBP needs $O(n^4 + n^2 L)$ time in total.

It remains to deal with the hardest case, where $k \geq 2$. We use that if $1 \leq i < k$ and $b_i < j < a_{i+1}$ then iteration j is fast. Denoting the number of these fast iterations as T_i, their iteration numbers are $b_i + 1$ to $b_i + T_i$. So one has $T_i = a_{i+1} - b_i - 1 \geq 1$.

LEMMA 6.3 *For each i such that $1 \leq i < k$ one has*

$$\frac{1}{\|z_{b_i}\|^2} - \frac{1}{\|z_{a_{i+1}}\|^2} < 4n^2\, O(T_i).$$

Proof To clarify the reasoning in this proof we include Table 1.

Since iteration b_i is slow it yields a cutting vector y and a non-cutting vector $\bar{y}$ as output, as well as the set J of indices for which y yields cuts. Then the MMA replaces y by $D\bar{y}/\mathbf{1}^{\mathrm{T}}\bar{y}$, where D is the diagonal matrix with $D_{ii} = \frac{1}{2}$ if $i \in J$ and $D_{ii} = 1$ if $i \notin J$. This vector is the input for MBP iteration $b_i + 1$ and denoted as $\tilde{y}_1$ in Table 1. Since iteration $b_i + 1$ is fast, $\tilde{y}_1$ is cutting. So it does not change $\tilde{y}_1$, but yields cuts according to the corresponding set J_1 in its output. Then the MMA changes $\tilde{y}_1$ to $\tilde{y}_2$ by rescaling the J_1-coordinates of $\tilde{y}_1$ and then normalizing, and so on.

As a result, after iteration $b_i + T_i$ the vector $\tilde{y}_{T_i}$ has the form $D\bar{y}/\mathbf{1}^{\mathrm{T}}D\bar{y}$, where the matrix D is the product of matrices D_j $(1 \leq j \leq T_i)$. This vector is the input at iteration a_{i+1}. Each D_j is a diagonal matrix with $|J_j|$ entries equal to $\frac{1}{2}$ and the remaining entries 1. It follows that D is a diagonal matrix with a most $\sum_{j=1}^{T_i} |J_j|$ entries less than or equal to $\frac{1}{2}$ and the remaining entries 1. Since $|J_j| = O(1)$, for each j, the number of entries less than 1 in D is $O(T_i)$.

Now Lemma 6.1 implies that

$$\frac{1}{\|z_{b_i}\|^2} - \frac{1}{\|z_{a_{i+1}}\|^2} < 2n^2\, O(T_i).$$

This proves the lemma. ■

Table 1. A sequence of MBP-iterations as considered in Lemma 6.3.

iteration number	b_i	b_i+1	b_i+2	.....	b_i+T_i	$b_i+T_i+1=a_{i+1}$
type	slow	fast	fast	.....	fast	slow
input		$\tilde{y}_1$	$\tilde{y}_2$	.....	$\tilde{y}_{T_i}$	
output	$y, \bar{y} \neq 0$	J_1	J_2	.....	J_{T_i}	

Table 2. Comparison of the modified method of Chubanov with Gurobi.

size(A)		iterations		accuracy	sizes K and J		time (s)	
m	n	MMA	MBP	$\|Ax\|$	$\|K\|$	$\|J\|$	Chubanov	Gurobi
5	10	2.0	2.5	9.7e − 15	0.2	5.0	0.0004	0.0011
25	50	3.5	47.0	1.8e − 13	5.6	17.9	0.0016	0.0020
125	250	4.7	918.0	3.5e − 12	37.3	69.7	0.0490	0.0303
625	1250	7.3	4676.2	2.3e − 11	243.5	525.3	4.7700	5.2611

Substitution of the inequality in Lemma 6.3 into (37) yields

$$N_s \leq n^3 + 2n^2 \sum_{i=1}^{k-1} O(T_i) \leq n^3 + 2n^2\, O(T).$$

Finally, using $T = O(nL)$ again we get

$$N = O(nL) + n^3 + 2n^2\, O(nL) = O(n^3 L).$$

Each MBP-iteration requires $O(n)$ time. Hence the contribution of the MBP to the time complexity is $O(n^4 L)$. As we established in (36) the contribution of the MMA is $O(n^3 L)$. Hence without further proof we may state our main result.

THEOREM 6.4 *The total time complexity of the MMA is* $O\left(n^4 L\right)$.

7. Computational results

To compare our approach with other approaches for solving linear systems we produced Table 2. Each line gives the average results for a class of 100 randomly generated problems with matrices A of size $m \times n$ as given in the first two columns. The elements of A were randomly chosen integers in the interval $[-100, 100]$, and uniformly distributed. For each of the given sizes the corresponding line gives the average number of iterations of the MMA and the MBP, the average accuracy and the average sizes of the sets K and J. The last two columns give the average solution times for our approach and for Gurobi, which is one of the fastest solvers nowadays, if not the fastest. Like any solver for LO problems, Gurobi cannot handle strict inequalities. So we used Gurobi with as input the following LO problem, which is equivalent to the homogeneous problem that we want to solve:

$$\min\{0^{\mathrm{T}} x \,:\, Ax = 0,\, x \geq e\}.$$

Taking into account that our implementation was in Matlab, and rather straightforward, it seems promising that the new approach competes with Gurobi. It must be admitted, however, that our experiments were conducted on a limited class of dense randomly generated problems and not on sparse problems.

8. Conclusion

Though conducted on a limited class of dense problems our comparison with Gurobi, which is nowadays one of the fastest solvers for linear systems, yielded promising results.

It remains as a topic for further research to find out if more can be said on the behaviour of the sizes of the sets K and J. In [1] these sets are always singletons. Our experiments made clear that taking larger sets strongly affects the computational behaviour. In the theoretical analysis, however, we were unable to take advantage of this. It may be noted that it may happen that during a slow iteration of the MBP the vector z has always precisely one negative entry. In that case the set K will be a singleton in each iteration, just as in [1]. However, since z is the orthogonal projection of a positive vector into the null space of a changing matrix, one might expect that this will be a rare event, as was confirmed during our experiments. On the other hand, if one, for example, could show that on average the size of K is a certain fixed fraction of the dimension n, this might open the way to further improvement of the iteration bound for the MMA.

Finally, as mentioned in [1], Chubanov's BP resembles a procedure proposed by Von Neumann that has quite recently been described by Dantzig [2]. This Von Neumann algorithm has been elaborated further in [4] and [8]. More recently it has been shown that the idea developed in the current paper can also be used to speed up Von Neumann's procedure [9]. Moreover, some authors successfully generalized Chubanov's method from LO to conic optimization (see, e.g. [7,10,11]).

Acknowledgements

Thanks are due to Guoyong Gu (Nanjing University) for pointing out a weakness in the proof of Lemma 4.2 in a previous version and to Sergei Chubanov for some helpful discussions on his paper [1]. Thanks are also due to Laszlo Végh and Giacomo Zambelli (London School of Economics) for pointing out errors in previous versions of the proof of Lemma 4.2.

Disclosure statement

No potential conflict of interest was reported by the author.

Notes

1. It can be shown that (21) is feasible if and only if $\|v\|^2 \leq \max(v)$. For a proof we refer to the appendix.
2. Problem (23) is the Lagrange dual of problem (22). Both problems have the same optimal value. In this proof we need only (24), which expresses the so-called *weak duality* property of the Lagrange dual.

References

[1] S. Chubanov, *A polynomial projection algorithm for linear feasibility problems*, Math. Program. 153 (2015), pp. 687–713. doi:10.1007/s10107-014-0823-8.
[2] G.B. Dantzig, *An ϵ-precise feasible solution to a linear program with a convexity constraint in $1/\epsilon^2$ iterations, independent of problem size*, Technical Report SOL 92-5, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, USA, October 1992.
[3] J. Edmonds, *Systems of distinct representatives and linear algebra*, J. Res. Natl. Bur. Stand. Sect. B 71B (1967), pp. 241–245.
[4] M. Epelman and R.M. Freund, *Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system*, Math. Program. 88(3, Ser. A) (2000), pp. 451–485.
[5] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, 1989.
[6] L.G. Khachiyan, *A polynomial algorithm in linear programming*, Doklady Akademiia Nauk SSSR 244 (1979), pp. 1093–1096. Translated into English in *Soviet Mathematics Doklady* 20, 191–194.
[7] T. Kitahara and T. Tsuchiya, *An extension of Chubanov polynomial-time linear programming algorithm to second-order cone programming*, 2016. Manuscript.
[8] D. Li and T. Terlaky, *The duality between the perception algorithm and the Von Neumann algorithm*, in *Modelling and Optimization: Theory and Applications*, L. Zukuaga and T. Terlaky, eds., Springer Science + Business, New York, 2013, pp. 113–136.
[9] D. Li, C. Roos, and T. Terlaky, *A polynomial column-wise rescaling von Neumann algorithm*,. Technical report 15T-010, Lehigh University, Bethlehem, USA, July 2015.

[10] B.F. Lourenco, T. Kitahara, M. Muramatsu, and T. Tsuchiya, *An extension of Chubanov's algorithm to symmetric cones*, 2017. Manuscript.
[11] J. Peña and N. Soheili, *Solving conic systems via projection and rescaling*, 2016. Manuscript.
[12] J. Renegar, *A polynomial-time algorithm, based on Newton's method, for linear programming*, Math. Program. 40 (1988), pp. 59–93.
[13] C. Roos, *On Chubanov's method for solving a homogeneous inequality system*, in *Numerical Analysis and Optimization. NAO-III, Muscat, Oman, January 2014*, M. Al-Baali, L. Grandinetti, and A. Purnama, eds., 2015, Proceedings in Mathematics & Statistics. ISBN 978-3-319-17688-8, Springer, Switzerland, pp. 319–338.
[14] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, New York, 1986.
[15] E. Stiemke, *Über positive Lösungen homogener linearer Gleichungen*, Math. Ann. 76 (1915), pp. 340–342.

Appendix. Feasibility condition for system (21)

LEMMA A.1 *Given $v \neq 0$, there exist y and z satisfying* (21) *if and only if* $\|v\|^2 \leq \max(v)$.

Proof Let $v \neq 0$. Suppose that y and z satisfy (21). Then $z^{\mathrm{T}}v = 0$ and $y = z + v$ imply $y^{\mathrm{T}}v = \|v\|^2$. One has

$$\max_y \{y^{\mathrm{T}}v \,:\, \mathbf{1}^{\mathrm{T}}y = 1,\, y \geq 0\} = \max(v),$$

which is attained if y is the unit vector e_i with i such that $v_i = \max(v)$. Hence it follows that $\|v\|^2 \leq \max(v)$. On the other hand, if $\|v\|^2 \leq \max(v)$ we need to show that there exist y and z that satisfy (21). Let $\lambda := \max(v)/\|v\|^2$. Then $\lambda\|v\|^2 = \max(v)$. Take $y = e_i$, with i such that $v_i = \max(v)$, and $z = y - \lambda v$. Then

$$z^{\mathrm{T}}v = (y - \lambda v)^{\mathrm{T}}v = y^{\mathrm{T}}v - \lambda v^{\mathrm{T}}v = v_i - \lambda\|v\|^2 = v_i - \max v = 0.$$

This proves the lemma. ■