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

Kees Roos, Delft University of Technology, c.roos@tudelft.nl

July 9, 2016

Abstract

We deal with a recently proposed method of Chubanov [1] 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

Contents

1 Introduction

Let A be an integer (or rational) matrix of size $m \times n$ and rank $(A) = m$. Recently Chubanov [1] presented a new algorithm that finds in polynomial time a solution of the system

$$Ax = 0,\ x > 0, \tag{1}$$

or establishes that no such solution exists. In the algorithm the author uses a nonzero vector $y \geq 0$ that is updated in each iteration and eventually serves to decide which of the two cases occurs. If a solution exists, then such a solution can be obtained from y.

A crucial tool in Chubanov's approach is a result showing that as long as no solution of (1) has been found, a 'better' y can be constructed. Eventually this leads to a 'small' vector y, which induces a 'cut' of the form $x_k \leq \frac{1}{2}$ for some index k.

In hindsight, duality theory helps to understand the role of the vector y in Chubanov's approach. For this we recall a variant of Farkas's lemma that is due to Stiemke [12]. It states that (1) has no solution if and only if the system

$$A^T u \geq 0, \quad A^T u \neq 0 \tag{2}$$

has a solution. Now one has $y = A^T u$ for some u if and only if $P_A y = 0$, where P_A denotes the orthogonal projection onto the null space of A. It follows that system (2) has a solution if and only if the system

$$P_A y = 0, \quad y \geq 0, \quad y \neq 0 \tag{3}$$

has a solution. Chubanov's algorithm can be viewed as a systematic search method for a vector y satisfying (3). It will be convenient to call any such vector a *dual feasible vector*.

Since (3) is homogeneous in y and $y \neq 0$, we may restrict the search to vectors y such that $\mathbf{1}^T y = 1$, 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*.

On the other hand, if y is not primal feasible then there must exist an index k such that $z_k \leq 0$. In that case it becomes natural to look for a new y' such that $\|P_A y'\| < \|P_A y\|$. This is exactly what the so-called Basic Procedure (BP) of Chubanov does, and [1, Lemma 2.1] shows how such an y' can be found.

Of course, if (1) has a positive solution then there is no y satisfying (3). A clever finding of Chubanov is to stop the BP when a y has been found such that

$$2\sqrt{n}\,\|P_A y\| \leq \max(y), \quad 0 \neq y \geq 0, \tag{4}$$

where $\max(y) := \max_i(y_i)$. In that case the vector y is said to be *small*. As Chubanov showed, this happens after at most $4n^3$ iterations, which makes his BP strongly polynomial. Any small vector y gives rise to a cut for problem (1) of the form $x_k \leq \frac{1}{2}$, where k is such that $y_k = \max(y)$. Hence it can be used to reduce problem (1) to a problem similar to (1), with A replaced by AD. Here D denotes the identity matrix I with I_{kk} replaced by $\frac{1}{2}$. Thus Chubanov's Main Algorithm (MA) in [1] replaces A by AD and then calls the BP again. If this yields a positive solution x for the new system, then Dx is a positive solution of (1); otherwise the BP will generate a new vector y satisfying (4), and so on.

Since A has integer (or rational) entries, the number of calls of the BP is polynomially bounded by the size of the matrix 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. As a result the algorithm solves problem (1) in polynomial time, namely in $O(nL)$ iterations of the Main Algorithm, where L denotes the bit size of A. This leads to an overall time complexity $O(n^5L)$. By performing a more careful analysis [1, Lemma 2.3] 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 [10, Section 1.2.2] we introduced a new cut criterion, but without further analysis, except that we made clear that the new cuts are at least as sharp as the cuts arising from (4). These cuts can be obtained in a much simpler way, as is shown in Section 2. 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., (13)) in Section 6. The stronger cut criterion reduces the iteration bound for the BP by a factor 5, without changing the order of its complexity. This is shown in Section 4, where we present our Modified Basic Procedure (abbr. MBP) and its analysis. A second improvement is in the search direction of the MBP. This search direction was also proposed [10]; in theory it performs much better than the search direction proposed by Chubanov, as was shown in [10, Lemma 5], which is Lemma 4.1 in the current paper. It has been acknowledged by Chubanov that our MBP speeds up the implementation of Chubanov's original version drastically [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

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

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

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

Note that our assumption $\text{rank}\,(A) = m$ implies that the inverse of AA^T exists. Obviously we have

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

Now let $y \in \mathbf{R}^n$, $y \geq 0$ and $\mathbf{1}^T y = 1$. In the sequel we use the notation

$$z = P_A y, \quad v = Q_A y.$$

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 'solves' the* dual *problem (2).*

Proof: The first statement has been established earlier. The second statement follows by noting that $v \in \mathcal{R}_A$ implies $v = A^T u$ for some u. Since A has full row rank, u is uniquely determined by v. □

The 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, i.e., $x' \in [0,1]^n$. It follows that (1) is feasible if and only if the system

$$Ax = 0, \;\; x \in (0,1]^n \tag{5}$$

is feasible. Moreover, if $d > 0$ is a vector such that $x \leq d$ holds for every feasible solution of (5) 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

$$ADx = 0, \;\; x \in (0,1]^n, \tag{6}$$

where $D = \mathrm{diag}\,(d)$. Obviously, problem (6) is of the same type as problem (5), since it arises from (5) by replacing A by 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 (5). However, while Chubanov used the vector z to construct cuts for (5), in this paper this is done by exploring properties of the vectors 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 $\mathbf{1}^T v^+$, and the sum of its negative entries by $\mathbf{1}^T v^-$. We call y is a *weak cutting vector* if

$$\mathbf{1}^T v^+ < -\min(v) \quad \text{or} \quad -\mathbf{1}^T v^- < \max(v). \tag{7}$$

The reason for this name is that if (7) holds then there exists at least one index k such that $x_k < 1$ holds for all solutions of (5). This is a consequence of the next lemma.

Lemma 2.2 *Let x be feasible for (5), $y \geq 0$ any nonnegative vector and $v = Q_A y$. Then every nonzero element v_k of v gives rise to a an upper bound for x_k, according to*

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

Proof: Since $Ax = 0$ we have $P_A x = x$. Hence $v^T x = v^T P_A x = x^T P_A v = 0$. Now suppose $v_k < 0$. Then we deduce from $v^T x = 0$ and $0 \leq x \leq \mathbf{1}$ that

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

where $\mathbf{1}$ denotes the all-one vector. On the other hand, if $v_k > 0$ we obtain in the same way

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

Hence we have

$$x_k \leq \begin{cases} \frac{\mathbf{1}^T v^+}{-v_k} & \text{if } v_k < 0, \\ \frac{\mathbf{1}^T v^-}{-v_k} & \text{if } v_k > 0. \end{cases} \tag{9}$$

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

Corollary 2.3 *If a nonzero entry v_k of v gives rise to a cut of the form $x_k \leq \tau < 1$ then v_k has the same sign as $\mathbf{1}^T v$.*

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

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

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

In the sequel we shall use only cuts of the form $x_k \leq \frac{1}{2}$. If the right-hand side expression in (8) does not exceed $\frac{1}{2}$ we call (8) 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 noncutting. 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. Below we give a simple example, where we use that the above conditions on v are homogeneous in v.

Example 2.4 *By way of example we consider the case where v (up to some positive factor) is given by*

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

Since $\mathbf{1}^T v = 13 > 0$ only positive entries v_k in v may give rise to a nonvoid cut, and this happens if v_k exceeds

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

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

$$x_1 \leq \frac{2}{3}, \quad x_2 \leq \frac{2}{4}, \quad x_6 \leq \frac{2}{6}.$$

3 More on cut-generating vectors

In Lemma 2.2 we showed how to obtain a cut $x_k \leq \frac{1}{2}$, for some k, for problem (5) 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

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

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

$$x_k \leq \mathbf{1}^T \left[e_k - \frac{v}{y_k}\right]^+. \tag{10}$$

One has

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

because $e_k - \frac{y}{y_k} \leq 0$. Hence we obtain

$$x_k \leq \mathbf{1}^T \left[\frac{z}{y_k}\right]^+ = \frac{\mathbf{1}^T z^+}{y_k} \leq \frac{\sqrt{n}\|z^+\|}{y_k} \leq \frac{\sqrt{n}\,\|z\|}{y_k}, \tag{11}$$

which is exactly the weaker cut used in [1].

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 $u = \frac{v}{y_k}$ 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

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

One may easily verify that $q(\alpha)$ is a nonnegative piecewise linear convex function with a breakpoint at $\alpha = 0$ and, if $v_k \neq 0$, another breakpoint at $\alpha = \frac{1}{v_k}$. 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 $\alpha = \frac{1}{v_k}$ 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 nonzero $y \geq 0$ and for each k one has

$$\min\left(1, \sum_{i=1}^n \left[\frac{-v}{v_k}\right]^+\right) \leq \mathbf{1}^T \left[e_k - \frac{v}{y_k}\right]^+ \leq \frac{\sqrt{n}\,\|z\|}{y_k}. \tag{12}$$

Since the left expression represents the minimal value of $q(\alpha)$, each of the three expressions is an upper bound for x_k. Of course, an upper bound is nonvoid if and only if its value is less than 1.

It may be worth noting that the expression in the middle yields a nonvoid upper bound only if $v_k > 0$. This easily follows because if $v_k \leq 0$ then the value of the k-th term alone in this expression already is at least 1. On the contrary, the left expression may yield a nonvoid cut also for negative entries of v.

Note that (12) implies a result that we will use later on, namely,

$$\sum_{i=1}^n \left[\frac{-v}{v_k}\right]^+ > \frac{1}{2} \quad \Rightarrow \quad y_k < 2\sqrt{n}\,\|z\|. \tag{13}$$

4 Modified Basic Procedure

In this section we show that if y is not primal or dual feasible then it is possible to find in $O(n^2)$ 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 (3);

(iii) y is a cutting vector.

In the first two cases the status of (1) is clear: in case *(i)* we have a solution of (1), 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 (5). Obvious such an inequality cuts off halve of the feasible region of (5). It enables us to update the current vector d by dividing its k-th 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 a positive vector y such that $\mathbf{1}^T y = 1$. The notation $\text{bound}_j(y)$ stands for the upper bound for x_j in Lemma 2.2. So

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

The smallest of these bounds is denoted as $\text{bound}(y)$. More precisely,

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

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

If the vector y is primal feasible or dual feasible and $\text{bound}(y) > \frac{1}{2}$ 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 z is negative or zero. Hence we may find a nonempty set K of indices such that

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

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

$$e_K := \frac{1}{|K|} \sum_{k \in K} e_k, \quad p_K := P_A e_K = \frac{1}{|K|} \sum_{k \in K} p^k. \tag{14}$$

Note that $0 \neq e_K \geq 0$, and $\mathbf{1}^T e_K = 1$. 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 $z^T p^k = z^T P_A e_k = z^T e_k = z_k$ for each k. Thus we obtain

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

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

```
INITIALIZE: z = P_A y; ȳ = 0; case = 0; J = ∅;
while bound(y) > 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 ≠ ∅ such that Σ_{k∈K} z_k ≤ 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
                    ȳ := y
                    α = p_K^T (p_K − z)/‖z − p_K‖^2
                    y = αy + (1 − α)e_K
                    z = αz + (1 − α)p_K  (= P_A y)
if case = 0 then
    find a nonempty set J such that J ⊆ {j : bound_j(y) ≤ 1/2}
```

As a consequence, in the equation

$$\|z - p_K\|^2 = \left(\|z\|^2 - z^T p_K\right) + \left(\|p_K\|^2 - z^T p_K\right) \tag{15}$$

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

$$\tilde{y} = \alpha y + (1-\alpha)e_K, \quad \alpha = \frac{\|p_K\|^2 - z^T p_K}{\|z - p_K\|^2} = \frac{p_K^T(p_K - z)}{\|z - p_K\|^2}. \tag{16}$$

Because of (15), 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 $\mathbf{1}^T y = \mathbf{1}^T e_K = 1$, also $\mathbf{1}^T \tilde{y} = 1$.

The transformation (16) from y to $\tilde{y}$ is the key element in Algorithm 1. It iterates (16) 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 $\frac{1}{\|\tilde{z}\|^2}$.

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 (16) and $\tilde{z} := P_A \tilde{y}$, one has*

$$\frac{1}{\|\tilde{z}\|^2} \geq \frac{1}{\|z\|^2} + |K|\,. \tag{17}$$

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,

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

The value of α that minimizes this expression is given in (16). It follows that

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

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

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

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 $\text{bound}\,(y) \geq \frac{1}{\sigma}$ for some $\sigma \geq 1$. Then we have $\text{bound}_k(y) \geq \frac{1}{\sigma}$ for all k such that $v_k \neq 0$. This means that

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

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 (nonzero) vectors in $\mathbf{R}^n$ that satisfy (19) is denoted as V. So we have

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

As a consequence, we have $v \in V$ if and only if the following two inequalities are satisfied:

$$-\sigma \mathbf{1}^T v^- \ \geq \ \max(v), \tag{20}$$
$$\sigma \mathbf{1}^T v^+ \ \geq \ -\min(v). \tag{21}$$

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

$$y \geq 0,\ \mathbf{1}^T y = 1,\ y = z + v,\ z^T v = 0, \tag{22}$$

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

$$\min_{y,z,\beta} \left\{ \|z\| \ : \ y \geq 0,\ \mathbf{1}^T y = 1,\ y = z + \beta v,\ z^T v = 0 \right\}. \tag{23}$$

We introduced an additional variable β because if $\beta = 1$ problem (23) may be infeasible.[1] An crucial observation is that if $\beta \neq 0$ then $v \in V$ if and only if $\beta v \in V$. Another important fact is that the problem is easy to solve if $\beta = 0$, because then $z = y$. Since $y \geq 0$ and $\mathbf{1}^T y = 1$ we then have $\|z\| \geq 1/\sqrt{n}$. The main result in this section is the following lemma, whose proof will make clear that smaller values of $\|z\|$ are achieved if $\beta \neq 0$.

Lemma 4.2 *Let $n \geq 2$, $\sigma \geq 2$ and $v \in V$. If y and z satisfy (22) then*

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

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

$$\max_{\alpha,\,\lambda} \left\{ \alpha \ : \ \lambda \geq \alpha \mathbf{1},\ \|\lambda\| \leq 1,\ \lambda^T v = 0 \right\}. \tag{24}$$

The relevance of this problem for our purpose is that if (y, z, β) is feasible for (23) and (λ, α) for (24), then one has

$$\|z\| \geq \|z\|\,\|\lambda\| \geq \lambda^T z = \lambda^T (y - \beta v) = \lambda^T y \geq \alpha \mathbf{1}^T y = \alpha, \tag{25}$$

where we used the Cauchy-Schwarz inequality and the feasibility conditions for both problems. This implies that if for some α there exist a feasible solution of (24) with the objective value $\geq \alpha$, for every $v \in V$, then we have $\|z\| \geq \alpha$ for every feasible solution of (23) with $v \in V$. This is the main argument in the rest of the proof.

Let $v \in V$. We define index sets S and T as follows:

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

[1] It can be shown that (22) is feasible if and only if $\|v\|^2 \leq \max(v)$. For a proof we refer to Appendix A.

[2] Problem (24) is the Lagrange dual of problem (23). Both problems have the same optimal value. In this proof we need only (25), which expresses the so-called *weak duality* property of the Lagrange dual.

With $\tau \geq \alpha \geq 0$, we consider the vector $\lambda \in \mathbf{R}^n$ defined by

$$\lambda = \left[\ \alpha \mathbf{1}_S;\ \tau \mathbf{1}_T\ \right], \tag{26}$$

where $\mathbf{1}_S$ and $\mathbf{1}_T$ denote the restrictions of the all-one vector $\mathbf{1}$ to the index sets S and T, respectively, and similar for other vectors. Then $\lambda \geq \alpha \mathbf{1}$ is feasible for (24) with objective value α if $\lambda^T v = 0$ and $\|\lambda\|^2 = 1$. In other words, α and τ must be such that

$$\begin{aligned} \alpha \mathbf{1}_S^T v_S + \tau \mathbf{1}_T^T v_T &= 0, & (27)\\ \alpha^2 |S| + \tau^2 |T| &= 1. & (28) \end{aligned}$$

Note that $v_S > 0$, whence $\mathbf{1}_S^T v_S > 0$. Similarly $\mathbf{1}_T^T v_T < 0$. The above system determines τ and α as follows:

$$\tau = \frac{\alpha \mathbf{1}_S^T v_S}{-\mathbf{1}_T^T v_T}, \quad \frac{1}{\alpha^2} = |S| + \frac{(\mathbf{1}_S^T v_S)^2}{(\mathbf{1}_T^T v_T)^2} |T|\,. \tag{29}$$

Using (20) and $v_T = v^-$ we obtain

$$\mathbf{1}_S^T v_S \leq |S| \max(v) \leq -|S|\, \sigma \mathbf{1}^T v^- = -|S|\, \sigma \mathbf{1}_T^T v_T.$$

Substitution into (29) yields

$$\frac{1}{\alpha^2} \leq |S| + \sigma^2 |S|^2 |T|\,. \tag{30}$$

It remains to find out how large the last expression can be. This question can be answered because of the equality $|S| + |T| = n$. Hence, putting $s = |S|$ and $|T| = n - s$ we need to find the maximal value of the function

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

One easily verfies that the largest value occurs if $s = \theta$, with $\theta = \frac{m + \sqrt{3+m^2}}{3\sigma}$, 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 $\frac{18}{25}m^2 + 15m + \frac{54}{25} \leq 1.4m^3$. Substitution gives

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

This implies the inequality in the lemma. □

Our main interest is the case where $\sigma = 2$. Then Lemma 4.2 yields that

$$\frac{1}{\|z\|^2} \leq \frac{4n^3}{5} < n^3. \tag{31}$$

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: As before, we assume that $\mathbf{1}^T y = 1$, $y > 0$ and $z = P_A y$. If $\text{bound}\,(y) \leq \frac{1}{2}$ then the MBP requires only 1 iteration. Otherwise $\text{bound}\,(y) > \frac{1}{2}$, which implies (31). 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 $\text{bound}_j(y)$ 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 noncutting 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 (5). 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 (6) in the following way:

$$Ax = 0, \quad 0 < x_i \leq d_i, \quad 1 \leq i \leq n, \tag{32}$$

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 (5) are bounded from below by τ [6, 9, 11].

Since the basic feasible solutions also satisfy $x \leq d$, we conclude that (5), and hence our 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 (5) 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

Algorithm 2: $[x, y, d, \text{case}] = \text{Modified Main Algorithm}(A, \tau)$

```
1: Initialize: d = e; y = e/n; x = 0; case = 0;
2: while case = 0 do
3:     P_A = I − A^T (A A^T)^{−1} A
4:     [y, ȳ, z, J, case] = Modified Basic Procedure(P_A, y)
5:     if case = 0 then
6:         d_J = d_J / 2
7:         if min(d_J) < τ then
8:             case = 3
9:         else
10:            if ȳ ≠ 0 then
11:                y = ȳ
12:            A_J = A_J / 2
13:            y_J = y_J / 2
14:            y = y / 1^T y
15: if case = 1 then
16:     D = diag(d)
17:     x = Dz
```

entries of d indexed by the set J by 2 and then checks if one 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 5.2, 5.2 and 5.3. As mentioned before this vector equals the zero vector if the MBP did not change the vector y, but otherwise it is the last noncutting vector generated by the MBP. The current y – which is a cutting vector with respect to the current A – is replaced by the noncutting 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

$$T \leq 1 + \sum_{i=1}^{n} \log_2 \frac{1}{\tau} = 1 + n \log_2 \frac{1}{\tau} = O(nL). \tag{33}$$

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

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, 9]. 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]

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

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

$$O(n^3) + O(n^2)O(nL) = O(n^3 L) \tag{34}$$

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

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 $O(n^4 L)$.

Crucial for our result is the next lemma. It slightly improves [1, Lemma 2.3]; in essence its proof differs from the proof of that lemma only in the last lines.

Lemma 6.1 *Let $\bar{y}$ be noncutting with respect to A and $D = \operatorname{diag}(d)$, with $0 \leq d \leq e$. Moreover, let $y = D\bar{y}/\mathbf{1}^T D\bar{y}$. If $\bar{z} = P_A \bar{y}$ and $z = P_{AD} y$, then*

$$\frac{1}{\|\bar{z}\|^2} - \frac{1}{\|z\|^2} < 2n^2 \left|I\right|,$$

where $I = \{i \ : \ d_i < 1\}$.

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 $\bar{v} = A^T u$ for some u. Using $P_{AD} D A^T = 0$ 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 $0 \leq d \leq e$ 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

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

Since $\mathbf{1}^T\bar{y} = 1$, and $d_i = 1$ if $i \notin I$, we may write

$$\mathbf{1}^T D\bar{y} = \sum_{i=1}^{n} d_i\bar{y}_i \geq \sum_{i\notin I} d_i\bar{y}_i = \sum_{i\notin I}\bar{y}_i = 1 - \sum_{i\in I}\bar{y}_i \geq 0.$$

We therefore obtain

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

Since $\bar{y}$ is noncutting, we derive from (13) that

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

Hence we may write, using $\frac{1}{\|\bar{z}\|} \leq n\sqrt{n}$, by (31),

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

This proves the lemma. □

We proceed by deriving an improved upper bound for the total number of MBP-iterations. We distinguish two types of MBP-iterations: iterations where the MBP changes y and iterations that leave y unchanged. Following Chubanov [1], we call these MBP-iterations *fast* and *slow*, respectively. The number of fast iterations is denoted as F and the number of slow iterations as S. Then the total number of MBP-iterations equals $N := F + S$.

It is obvious that an MBP-iteration is slow if and only if y is not primal or dual feasible and noncutting. On the other hand, a fast iteration occurs if and only if at the start of the while loop in the MBP y is primal feasible or dual feasible or a proper cutting vector.

We number the MBP iterations from 1 to N. We denote the y- and z-vector at the start of the while loop at iteration j as y_j and z_j, respectively. Furthermore, the indices $a_1, \ldots a_k$ and $b_1, \ldots b_k$ are defined such that $a_i \leq b_i < a_{i+1}$ for $1 \leq i < k$ and such that if $a_i \leq j \leq b_i$ 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 fast MBP-iterations. Note that fast iterations may also occur before the first train and after the last train.

It may happen that $k = 0$, i.e, all MBP iterations are fast. Then $F = T = O(nL)$, and the MBP needs $O(n^2L)$ time in total. So we assume below that $k \geq 1$. Then we have the following result.

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

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

Proof: One has

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

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

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

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

By rearranging the terms in the above sum we obtain

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

Since iteration b_k is slow, the y-vector at the start of the while loop is noncutting. Due to (31) this implies that the first term in the last expression does not exceed n^3. Neglecting the second term we obtain

$$S \leq n^3 + \sum_{i=1}^{k-1} \left(\frac{1}{\|z_{b_i}\|^2} - \frac{1}{\|z_{a_{i+1}}\|^2} \right). \tag{35}$$

If $k = 1$, this gives $S \leq n^3$, in accordance with Theorem 4.3. Then $N \leq n^3 + F \leq n^3 + T$, 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. The number of these iterations is denoted as F_i. So one has $F_i = a_{i+1} - b_i - 1 \geq 1$.

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

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

Proof: To clarify the reasoning in this proof we include Table 1. Since iteration b_i is

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

iteration number	b_i	$b_i + 1$	$b_i + 2$	.....	$b_i + F_i$	$b_i + F_i + 1 = a_{i+1}$
type	slow	fast	fast	.....	fast	slow
input	$\bar{y}$	y	$D_1\bar{y}$	.....	$D_{F_i-1}\bar{y}$	$D_{F_i}\bar{y}$
output	$y, \bar{y}$	$J \neq \emptyset$	$J \neq \emptyset$	.....	$J \neq \emptyset$	??

slow and iteration $b_i + 1$ is fast, iteration $b_i + 1$ starts with the proper cutting vector y that was generated during iteration b_i and yields an index set J for the cuts induced by y, without changing the noncutting vector $\bar{y}$ that was the input for iteration b_i. After this the

MMA calls the remaining $F_i - 1$ fast iterations, without changing $\bar{y}$. After each of the F_i fast iterations the MMA multiplies $O(1)$ entries in d by $\frac{1}{2}$. The resulting D-matrices are denoted as $D_1, \ldots D_{F_i}$. Hence at the start of iteration $b_i + F_i + 1$ the y-vector is given by $y = D\bar{y}/\mathbf{1}^T D\bar{y}$ where $D = D_{F_i}$ and where $d = \text{diag}\,(D)$ has at most $O(F_i)$ entries less than 1. Now Lemma 6.1 implies that

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

This proves the lemma. □

Substitution of the inequality in Lemma 6.3 into (35) yields

$$S \leq n^3 + 2n^2 \sum_{i=1}^{k-1} O(F_i) \leq n^3 + 2n^2\, O(F).$$

One easily verifies that $F = T$, where T is the total number of MMA-iterations. Since the total number of MBP-iterations equal $N = F + S$, we obtain

$$N \leq T + n^3 + 2n^2\, O(T).$$

Finally, using $T = O(nL)$, by (33), 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 (34) 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:

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

Taking into account that our implementation was in Matlab, and rather straightforward, it seems promising that the new approach competes with Gurobi.

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

size(A)		iterations		accuracy	sizes K and J		time (sec)	
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

8 Conclusion

From our numerical experiments we conclude that Chubanov's MA, when equipped with the (new) MBP is competitive with Gurobi, nowadays one of the fastest solvers for linear systems of the form studied in this paper.

It remains as a topic for further research to find out if more can be said on the behavior 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 behavior. 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, e.g., 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 in [2]. This Von Neumann algorithm has been elaborated further in [4] and [7]. More recently it has been shown that the results of this paper can be used to get a polynomial time version of Von Neumann's procedure [8].

Acknowledgement

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.

A Feasibility condition for system (22)

Lemma A.1 *Given $v \neq 0$, there exist y and z satisfying (22) if and only if $\|v\|^2 \leq \max(v)$.*

Proof: Let $v \neq 0$. Suppose that y and z satisfy (22). Then $z^T v = 0$ and $y = z + v$ imply $y^T v = \|v\|^2$. One has

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

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 (22). 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

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

This proves the lemma. □

References

[1] Sergei Chubanov. A polynomial projection algorithm for linear feasibility problems. *Mathematical Programming*, 153:687–713, 2015. 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. Nat. Bur. Standards Sect. B*, 71B:241–245, 1967.

[4] Marina Epelman and Robert 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):451–485, 2000.

[5] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, second edition, 1989.

[6] L. G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademiia Nauk SSSR*, 244:1093–1096, 1979. Translated into English in *Soviet Mathematics Doklady* 20, 191–194.

[7] D. Li and T. Terlaky. The duality between the percepton algorithm and the Von Neumann algorithm. In L. Zukuaga and T. Terlaky, editors, *Modelling and Optimization: Theory and Applications*, pages 113–136, Springer Science+Business, New York, 2013.

[8] Dan Li, Kees Roos, and Tamás Terlaky. A polynomial column-wise rescaling von Neumann algorithm. Technical report 15T-010, Lehigh University, Bethlehem, USA, July 2015.

[9] J. Renegar. A polynomial-time algorithm, based on Newton's method, for linear programming. *Mathematical Programming*, 40:59–93, 1988.

[10] Kees Roos. On Chubanovs method for solving a homogeneous inequality system. In Mehiddin Al-Baali, Lucio Grandinetti, and Anton Purnama, editors, *Numerical Analysis and Optimization. NAO-III, Muscat, Oman, January 2014*, pages 319 – 338, Switzerland, 2015. Springer Proceedings in Mathematics & Statistics. ISBN 978-3-319-17688-8.

[11] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, New York, 1986.

[12] E. Stiemke. Über positive Lösungen homogener linearer Gleichungen. *Mathematische Annalen*, 76:340–342, 1915.