A MODIFIED ALGORITHM FOR THE STRICT FEASIBILITY PROBLEM

D. BENTERKI\textsuperscript{1} AND B. MERIKHI\textsuperscript{1}

Communicated by Jean-Pierre Crouzeix

Abstract. In this note, we present a slight modification of an algorithm for the strict feasibility problem. This modification reduces the number of iterations.

Keywords: Strict feasibility, interior point methods, Ye–Lustig algorithm.

1. INTRODUCTION

This paper is concerned with the problem of finding $x \in \mathbb{R}^n$ such that

$$x > 0 \quad \text{and} \quad Ax = b$$

where $A$ is a $m \times n$ real matrix of rank $m$, $b \in \mathbb{R}^m$ and $0 < m < n$.

This problem, called a strict feasibility problem, occurs in many optimization problems in linear or quadratic programming. Such problems are of type

Minimize $f(x)$ subject to $x \in S = \{x \in \mathbb{R}^n : x \geq 0, Ax = b\}$. \hspace{1cm} (P)

Let us denote by $\widetilde{S}$ the following subset:

$$\widetilde{S} = \{x \in \mathbb{R}^n : x > 0, Ax = b\}.$$
Interior point methods for solving \((P)\) start from some arbitrary initial point \(x^0 \in \mathcal{S}\) and build a sequence \(\{x^k\} \subset \mathcal{S}\) expected to converge to some optimal solution \(x^*\) of \((P)\). Thus, the first step in interior point methods consists in finding an initial point \(x^0\) in an efficient way. To do that, it is usual to introduce an artificial variable \(\lambda\) and to consider the linear programming problem

\[
\text{Minimize}_{x, \lambda} \{ \lambda : Ax + \lambda(b - Aa^0) = b, \ x \geq 0, \ \lambda \geq 0 \} \quad (LF)
\]

where \(a^0\) is an arbitrary fixed point in the positive orthant of \(\mathbb{R}^n\). It is noticed that \(x^*\) is a solution of the feasibility problem \((F)\) if and only if \((x^*, 0)\) is an optimal solution of \((LF)\) with \(x^* > 0\).

In fact, numerical algorithms provide only approximate optimal solutions for an optimization problem. In our case, an approximate solution of \((LF)\) can be obtained via a classical interior method as for instance the Ye–Lustig algorithm that we described below. But before, we precise the notation used in this algorithm: \(\varepsilon > 0\) corresponds to the precision of the approximation, 

\[
r = \frac{1}{\sqrt{(n+1)(n+2)}}
\]

\(c = (0, 0, \cdots, 0, 1)^t \in \mathbb{R}^{n+1}\), \(e_{n+2} = (1, 1, \cdots, 1)^t \in \mathbb{R}^{n+2}\), \(\mathcal{F} = (x, \lambda)^t \in \mathbb{R}^{n+1}\) and 

\(B = [A, b - Aa^0]\) is a \(m \times (n + 1)\) matrix.

## 2. The original algorithm and its modification

Let us describe the original algorithm:

### The Ye–Lustig algorithm [3]

- **a)** Initialization: start with \(x^0 = a^0\), \(\lambda^0 = 1\), \(\bar{x}^0 = (x^0, \lambda^0)^t\) and \(k = 0\);
  - If \(\|Ax^0 - b\| \leq \varepsilon\) **Stop**: \(x^0\) is an \(\varepsilon\)-approximate solution,
  - If **not** go to b).
- **b)** If \(\lambda^k \leq \varepsilon\) **Stop**: \(x^k\) is an \(\varepsilon\)-approximate solution,
  - If **not** go to c).
- **c)** Set \(D^k = \text{diag}(\bar{x}^k)\) and
  - Compute the projection \(p^k\) of the vector \((D^k c, -c^t \bar{x}^k)^t \in \mathbb{R}^{n+2}\) on the kernel of the \(m \times (n + 2)\) matrix \(B^k = [BD^k, -b]\),
  - Take \(\gamma^{k+1} = \frac{\bar{x}^k}{\|\bar{x}^k\|} - \alpha^k r_{\frac{\gamma^k}{\|\gamma^k\|}}\), where \(\alpha^k\) is obtained by a line search,
  - Take \(\tilde{x}^{k+1} = (x^{k+1}, \lambda^{k+1})^t = (y_{n+2}^{k+1})^{-1} D^k y^{k+1} [n + 1]\),
- **d)** do \(k = k + 1\) and go back to b).

We propose a slight modification of this algorithm by modifying the stopping criteria.
The modified algorithm

- a') Initialization: start with $x^0 = a^0$, $\lambda^0 = 1$, and $k = 0$.
  - If $\|Ax^0 - b\| \leq \varepsilon$ Stop: $x^0$ is an $\varepsilon$-approximate solution.
  - If not compute the solution $u^0$ of the linear system $AA^t u^0 = \lambda^0 (b - Aa^0)$.

- b') If $\lambda^k \leq \varepsilon$ Stop: $x^k$ is an $\varepsilon$-approximate solution.
  - If not take $u^k = \lambda^k u^0$.
  - Take $z^k = -\text{diag}(x^k)^{-1} A^t u^k$.
  - If $\max |z^k|_i < 1$ Stop: $x^k + A^t u^k$ is an $\varepsilon$-approximate solution.
  - If not go to c')

- c') is identical to c) of the original algorithm.
- d') do $k = k + 1$ and go back to b').

The computation of the vector $u^0$ occurs once only, it can be performed by a Cholesky method. It remains to prove the validity of the new stopping criteria $\max |z^k|_i < 1$. This is done in the following proposition.

**Proposition 2.1.** If $\max |z^k|_i < 1$ then $x^k + A^t u^k > 0$ and $A(x^k + A^t u^k) = b$.

**Proof.** 1) Notice that $-\text{diag}(x^k) z^k = A^t u^k$, then $x^k + A^t u^k = x^k - \text{diag}(x^k) z^k = \text{diag}(x^k) (e_n - z^k) > 0$ because $x^k > 0$ and $|z^k|_i < 1$ for all $i$.

2) Since $(x^k, \lambda^k)^t$ is a feasible solution of (LF) then $A(x^k + A^t u^k) = Ax^k + AA^t u^k = b - \lambda^k (b - Aa^0) + \lambda^k AA^t u^0 = b - \lambda^k (b - Aa^0) + \lambda^k \lambda^0 (b - Aa^0) = b$.

This modification brings a significant improvement in the number of iterations with only a very small increasing in the cost per iteration. We illustrate that by a few examples.

3. Examples

In this examples $\varepsilon$ has been taken equal to $10^{-3}$ or $10^{-6}$ according to the case.

3.1. Some examples

The following examples are taken form the literature see for instance [1, 4]. In particular, Example 7 is called the Hitac problem.
3.2. Cube example

\( n = 2m, A[i,j] = 0 \) \( \text{if} \ i \neq j \ \text{or} \ (i+1) \neq j \)

\( A[i,i] = A[i,i+m] = 1, b[i] = 2, \text{for} \ i,j = 1 \cdots m. \)

<table>
<thead>
<tr>
<th>Dimension ( m \times n )</th>
<th>Nbr of iterations Ye–Lustig</th>
<th>Nbr of iterations Modified algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 \times 100</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>100 \times 200</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>150 \times 300</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>200 \times 400</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

3.3. Hilbert example

\( n = 2m, A[i,j] = \frac{1}{i+j}, A[i,i+m] = 1, \)

\( b[i] = \sum_{j=1}^{m} \frac{1}{i+j}, \text{for} \ i,j = 1 \cdots m. \)

<table>
<thead>
<tr>
<th>Dimension ( m \times n )</th>
<th>Nbr of iterations Ye–Lustig</th>
<th>Nbr of iterations Modified algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 \times 100</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>100 \times 200</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>150 \times 300</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>200 \times 400</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
References


to access this journal online:
www.edpsciences.org