sciendo

Műszaki Tudományos Közlemények vol. 15. (2021) 15–18.
DOI: Hungarian: https://doi.org/10.33895/mtk-2021.15.04
English: https://doi.org/10.33894/mtk-2021.15.04

EME MTSZ MTK

Implementation of the Full-Newton Step Algorithm for Weighted Linear Complementarity Problems

Zsolt DARVAY,[1] Attila-Szabolcs ORBÁN [2]

[1] *Babeș-Bolyai University, Faculty of Mathematics and Computer Science, Cluj-Napoca, Romania, darvay@cs.ubbcluj.ro, Transylvanian Museum Society, Department of Mathematics and Computer Science*

[2] *Babeș-Bolyai University, Faculty of Mathematics and Computer Science, Cluj-Napoca, Romania, orban.attila@yahoo.com*

Abstract

We present a path-following interior-point algorithm for solving the weighted linear complementarity problem from the implementation point of view. We studied two variants, which differ only in the method of updating the parameter which characterizes the central path. The implementation was done in the C++ programming language and the obtained numerical results prove the efficiency of the proposed method.

Keywords: *weighted linear complementarity problem, path-following interior-point algorithm, full-Newton step method.*

1. Introduction

Many optimization problems can be given as a linear complementarity problem, which can be denoted as LCP [1, 2]. The LCP has been shown to be useful for a variety of practical problems including engineering or economic issues.

We deal with an extended version of the LCPs, which has more practical applications than the previous ones [3–5]. This is the so-called weighted linear complementarity problem (WLCP), which was introduced by Florian Potra in 2012 [6].

In the following, we will describe the problem.

2. The description of the problem

The WLCP can be formulated as:

$$\begin{cases} xs = w, \\ -Mx + s = q, \\ x \geq 0, s \geq 0, \end{cases}$$

where $q \in \mathbb{R}^n, M \in \mathbb{R}^{n\times n}$, and for the weight vector we have $w \in \mathbb{R}^n_+ = \{x \in \mathbb{R}^n | x \geq 0\}$. Furthermore, we assume that M is a monotone matrix, so $x^T Mx \geq 0$ for any $x \in \mathbb{R}^n$. The matrix M and the vector q are given, and x, s are unknown vectors.

Interior-point methods (IPMs), especially primal-dual IPMs, have been proven very effective for solving LCPs. In the case of these algorithms, we assume that the set of feasible solutions of the WLCP, denoted by $\mathcal{F}$ is not empty:

$$\mathcal{F} := \{(x, s) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ : -Mx + s = q\}.$$

The set of optimal solutions of the WLCP consists of feasible (x, s) pairs for which the complementarity condition is also satisfied. The set obtained in this way is denoted by $\mathcal{F}^*$:

$$\mathcal{F}^* := \{(x, s) \in \mathcal{F} : xs = w\}.$$

Finally, if x and s are positive vectors, we obtain the set of strictly feasible solutions, denoted by $\mathcal{F}^0$:

$$\mathcal{F}^0 := \{(x, s) \in \mathcal{F} : x, s > 0\}.$$

In the following, we assume that we have a strictly feasible $(x^0, s^0) \in \mathcal{F}^0$ starting point and we assign to them the vector c and the constant μ^0:

$$c = x^0 s^0, \qquad \mu^0 = \frac{(x^0)^T s^0}{n}.$$

Starting from μ^0, the algorithm determines different values for μ, that characterize an ideal trajectory, the so-called central path. For conventional not weighted LCPs, the central path is defined by the following system:

$$\begin{cases} xs = \mu e, \\ -Mx + s = q, \\ x > 0, s > 0, \end{cases}$$

where e is the n-dimensional all-one vector.

For WLCP, the right-hand side vector of the nonlinear equation ($xs = \mu e$) is replaced by the following expression [6, 7]:

$$w(\mu) = \left(1 - \frac{\mu}{\mu^0}\right) w + \frac{\mu}{\mu^0} c, \qquad \mu \in [0, \mu^0]. \quad (1)$$

If $w = 0$ and $x^0 = s^0 = e$, we obtain the well-known case, so $w(\mu) = \mu e$. The definition of $w(\mu)$ is justified by the fact that we obtain the vector c for $\mu = \mu^0$ and the weight vector w in the case of $\mu = 0$.

In the following, we assume that the value of $w(\mu)$ is given by equation (1). Therefore, in this case, the central path is described by the following system:

$$\begin{cases} xs = w(\mu), \\ -Mx + s = q, \\ x > 0, s > 0. \end{cases}$$

Due to the nonlinear equation, we will not be able to determine exactly the points of the central path, but we can follow this trajectory using Newton's method:

$$\begin{cases} s\Delta x + x\Delta s = w(\mu) - xs, \\ -M\Delta x + \Delta s = 0. \end{cases}$$

The vectors Δx and Δs give so-called search directions, which allow us to determine the vectors x and s of the next iteration. In case of full-Newton step algorithms, the new points are given by the following equations:

$$x^+ = x + \Delta x, \qquad s^+ = s + \Delta s.$$

The distance of a given point from the central path can be estimated using the following proximity measure:

$$\delta(x, s; \mu) = \left\| \frac{w(\mu) - xs}{\mu} \right\|.$$

It is important to mention that X and S are diagonal matrices formed by the components of x and s, respectively:

$X = diag(x), \ S = diag(s).$

We modify the full-Newton step algorithm introduced in [7] from the implementation point of view.

3. Implementation of the algorithm

From the implementation perspective, we introduced two modified versions of the algorithm. These differ in the method of updating the parameter μ which will be expressed by the procedure *updateOfMu()* of the following algorithm.

Input parameters:
- threshold parameter $\tau \in (0, 1)$;
- accuracy parameter $\epsilon > 0$;
- parameter that modifies the value of μ, in the first case $\theta \in (0, 1)$ and in the second case $\sigma \in (0, 1]$;
- step length reduction parameter $\rho \in (0, 1)$;
- the matrix M and the vector q;
- the weight vector w.

Output: x, s

Let $(x^0, s^0) \in \mathcal{F}^0$ and $\mu^0 = \frac{(x^0)^T s^0}{n}$ so that $\delta(x^0, s^0; \mu^0) \le \tau$;

$x := x^0$; $s := s^0$; $\mu := \mu^0$;

while $\left(\frac{\|xs - w\|}{1 + \|c\|} > \epsilon\right)$ **or** $\left(\frac{\|Mx + q - s\|}{1 + \|q\|} > \epsilon\right)$ **do**

updateOfMu();

$w(\mu) = \left(1 - \frac{\mu}{\mu^0}\right) w + \frac{\mu}{\mu^0} c$;

$r_q = Mx + q - s$;

$rhs = w(\mu) - xs$;

$\Delta x = (M + X^{-1}S)^{-1}(X^{-1} rhs - r_q)$;

$\Delta s = X^{-1}(rhs - S\Delta x)$;

$\alpha_x = min\left\{-\frac{x_i}{\Delta x_i} | 1 \le i \le n, \Delta x_i < 0\right\}$;

$\alpha_s = min\left\{-\frac{s_i}{\Delta s_i} | 1 \le i \le n, \Delta s_i < 0\right\}$;

$\alpha = min\{\alpha_x, \alpha_s, 1\}$;

$x := x + \rho\alpha\Delta x$; $s := s + \rho\alpha\Delta s$;

end while

In the first version, similar to the theoretical algorithm, the value of the parameter μ is always reduced by a constant multiplier, which will be determined by the parameter θ.

procedure updateOfMu()
begin
$\mu := (1 - \theta)\, \mu$;
end.

However, in case of the implementation of IPMs, we usually proceed by determining the next value of μ based on the current vectors x and s. To do this, we would like to find a value of μ for which the equation $xs = w(\mu)$ is satisfied. By expressing the value of μ we get the following equation:

$$\mu = \frac{\mu_0 (x^T s - e^T w)}{e^T c - e^T w}. \quad (2)$$

Notice that in this case we have to assume that $e^T c \neq e^T w$. If this condition is not met, we have to choose other x^0 and s^0 starting points.

Observe that if $w = 0$ relation (2) can be given in the form:

$$\mu = \frac{x^T s}{n}$$

which can be used in case of solving many unweighted LCPs.

In the second version of the *updateOfMu*() procedure, the value of μ calculated as above is reduced by a factor $\sigma \in (0, 1]$:

procedure updateOfMu()
begin

$$\mu := \sigma \frac{\mu_0 (x^T s - e^T w)}{e^T c - e^T w};$$

end

The implementation was done in C++ programming language, using Visual Studio development environment. It is based on the code introduced in [8].

4. Numerical results

The algorithm was tested on positive semidefinite input matrices that were generated using the following method:

$M = LL^T$,

where L is a lower triangular matrix. In the first case, let:

$$L = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 9 & -4 & 1 & 0 \\ -2 & 1 & 7 & 3 \end{pmatrix}.$$

The matrix M corresponding to the 4×4 dimension input, the vector q and the weight vector w are:

$$M = \begin{pmatrix} 25 & 5 & 45 & -10 \\ 5 & 10 & -3 & 1 \\ 45 & -3 & 98 & -15 \\ -10 & 1 & -15 & 63 \end{pmatrix},$$

$q = -Me + e$,

$w = [0.5 \quad 1 \quad 15 \quad 0.3]^T$.

In addition to the problem described above, we analyzed three other cases that are presented on the webpage [9] and are related to matrices of 50×50, 100×100 and 700×700 dimensions. These matrices and the weight vectors assigned to the problems were generated randomly and the value of q was always determined using the equation $q = -Me + e$.

The results shown in Tables 1 and 2 correspond to the two versions of the algorithm. In the headers of the tables, the numbers indicate the dimension of the matrix. In all cases we used the parameters $\rho = 0.95$ and $\epsilon = 10^{-5}$.

It can be stated that increasing θ or decreasing σ results in less number of iterations in each case.

Table 1. *Results for* θ

θ	4	50	100	700
0.1	124	128	129	129
0.2	59	61	61	61
0.3	37	38	38	38
0.4	26	27	27	27
0.5	19	20	20	20
0.6	15	15	15	15
0.7	11	12	12	12
0.8	9	9	9	9
0.9	6	7	8	8

Table 2. *Results for* σ

σ	4	50	100	700
0.1	8	8	9	10
0.2	10	11	12	12
0.3	13	13	14	15
0.4	17	17	18	18
0.5	21	22	23	23
0.6	28	29	30	30
0.7	40	40	41	41
0.8	62	64	64	65
0.9	131	135	135	136

5. Conclusions

In this paper, we analyzed the WLCP from the implementation point of view. For that, we defined two versions of the algorithm.

Using the code implemented in C++ programming language, we proved the efficiency of the algorithm. In case of different positive semidefinite input matrices, we studied the modification of the number of iterations as a function of the parameters θ and σ, respectively.

Acknowledgement

The authors would like to thank the Transylvanian Museum Society for the support of this research.

References

[1] Cottle R. W., Pang J.-S., Stone R. E.: *The Linear Complementarity Problem.* Computer Science and Scientific Computing. Academic Press, Boston, 1992.

[2] Kojima M., Megiddo N., Noma T., Yoshise A.: *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems.* Springer Verlag, Berlin, Germany, 1991.

[3] Ye Y.: *A Path to the Arrow-Debreu Competitive Market Equilibrium.* Mathematical Programming volume 111/1–2. (2008) 315–348. https://doi.org/10.1007/s10107-006-0065-5

[4] Anstreicher K. M.: *Interior-Point Algorithms for a Generalization of Linear Programming and Weighted Centring, Optimization Methods and Software*. Optimization Methods and Software, 27/4–5. (2012) 605–612. https://doi.org/10.1080/10556788.2011.644791

[5] Jian Z.: *A Smoothing Newton Algorithm for Weighted Linear Complementarity Problem*. Optim Letters, 10. (2016) 499–509. https://doi.org/10.1007/s11590-015-0877-4

[6] Potra F. A.: *Weighted Complementarity Problems – a new paradigm for computing equilibria.* SIAM Journal on Optimization, 22/4. (2012) 1634–1654. https://doi.org/10.1137/110837310

[7] Asadi S., Darvay Zs., Lesaja G., Mahdavi-Amiri N., Potra F. A.: *A Full-Newton Step Interior-Point Method for Monotone Weighted Linear Complementarity Problems*. Journal of Optimization Theory and Applications, 186/3. (2020) 864–878. https://doi.org/10.1007/s10957-020-01728-4

[8] Darvay Zs., Takó I.: *Computational comparison of primal-dual algorithms based on a new software*. unpublished manuscript, 2012.

[9] Darvay Zs., Orbán A. Sz.: *Generated positive semidefinite matrices and weight vectors* (accessed on: 2021.03.06). http://cs.ubbcluj.ro/~darvay/semidefinite/