Műszaki Tudományos Közlemények 17. (2022) xx–yy.
Magyar nyelven: https://doi.org/10.33895/mtk-2022.17.xx
Angol nyelven: https://doi.org/10.33894/mtk-2022.17.xx

Algebraically equivalent transformation technique for weighted linear complementarity problems

Darvay Zsolt[1], Jakab Zsanett[2]

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

[2] *Babeș-Bolyai University, Faculty of Mathematics and Computer Science. Cluj-Napoca, Romania, zsanett28@yahoo.com*

Abstract

We study the algebraically equivalent transformation technique using an application implemented in the Java programming language that solves weighted linear complementarity problems. In general, in case of the algebraically equivalent transformation, we divide the nonlinear equation of the system that gives the central path, the so-called centrality equation, by the barrier parameter. However, in this case, we divide component-wise the centrality equation by the right-hand side vector, which depends on the barrier parameter. We apply the same continuously differentiable and invertible function to both sides of the obtained equation and generate different search directions using Newton's method. We analyze the numerical results provided by our application in case of applying the algebraically equivalent transformation technique using the identical map and the square root function.

Keywords: *weighted linear complementarity problem, algebraically equivalent transformation, interior-point algorithm.*

1. Description of the weighted linear complementarity problem

The weighted complementarity problem (*WLCP*) significantly extends the concept of the traditional linear complementarity problem (*LCP*). *WLCP* was introduced by Potra [1] in 2012.

Let $\mathbb{R}^n_+$ be the set of vectors consisting of nth order non-negative real numbers. Similar to the traditional complementarity problem, the following system can be considered [2]:

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

where $x, s, q \in \mathbb{R}^n$, $M \in \mathbb{R}^{n\times n}$ and $w \in \mathbb{R}^n_+$.

It can be observed that in the case of the weighted problem, the xs component wise multiplication must be equal to the vector w, consisting of non-negative real numbers. In addition, the relation $-Mx + s = q$ must be fulfilled, as well as the fact that the components of the vectors x and s are non-negative real numbers.

Interior-point algorithms usually follow the central path, but in the case of WLCP, we introduce a special path. Assuming that the initial x_0 and s_0 interior points are given and $\mu^0 = \frac{(x^0)^T s^0}{n}$, we introduce the following vector depending on the positive parameter μ [1, 2]:

$$w(\mu) = \left(1 - \frac{\mu}{\mu_0}\right) w + \frac{\mu}{\mu_0} c, \text{ where } c = x_0 s_0.$$

Thereby, the central path, which designates the direction, can be specified as follows:

$$\begin{aligned} -Mx + s &= q, \\ xs &= w(\mu), \\ x > 0, s &> 0, \end{aligned} \qquad (1)$$

where $\mu \in (0, \mu^0]$. In the following, we will introduce a possible modification of the above system.

2. Algebraically equivalent transformation technique

Let $e = [1\,1 \ldots 1]^T$ be an n-dimensional vector consisting of ones. Considering the algebraically equivalent transformation technique [3, 4], in the traditional case, we proceed by dividing the centrality equation $xs = \mu\, e$ by the barrier parameter μ, and then applying a continuously differentiable and invertible φ function to both sides of the equation. Thus, we obtain the equation $\varphi\left(\frac{xs}{\mu}\right) = \varphi(e)$. On the other hand, in the case of the WLCP, the centrality equation is given in the form of $xs = w(\mu)$, therefore we will divide it by component with the right-hand vector $w(\mu)$. Following this, we apply the function φ, so system (1) takes the following form:

$$\begin{aligned} &-Mx + s = q, \\ &\varphi\left(\frac{xs}{w(\mu)}\right) = \varphi(e), \\ &x > 0, s > 0. \end{aligned}$$

Using Newton's method, we get the system that already applies to the search directions Δx and Δs:

$$\begin{aligned} &s\Delta x + x\Delta s = w(\mu) \cdot \frac{\varphi(e) - \varphi\left(\frac{xs}{w(\mu)}\right)}{\varphi'\left(\frac{xs}{w(\mu)}\right)}, \\ &-M\Delta x + \Delta s = 0. \end{aligned} \qquad (2)$$

It can be established that by introducing the function φ, the matrix of the system of linear equations defining the search directions will always be the same, only the right-hand vector depends on the function φ. In the practical implementation, we use two types of φ functions, the identical map and the square root function: $\varphi(t) = t$, $\varphi(t) = \sqrt{t}$.

To analyze the algorithm, the function δ must also be introduced, which defines the environment of the given point of the central path. This gives us a proximity measure that expresses the distance of points x and s from the central path:

$$\delta(x, s, \mu) = \left\| \frac{w(\mu)}{\mu} - v^2 \right\|, \text{where } v = \sqrt{\frac{xs}{\mu}}, \mu > 0.$$

In the following, we will describe the modified version of the algorithm from the point of view of theory and implementation.

3. The algorithm

First, we present the full-step interior-point algorithm, and then we introduce a modified version from the point of view of implementation, which is suitable for the effective solution of weighted linear complementarity problems. The theoretical algorithm can be written as follows [2].

```
Theoretical algorithm for WLCP
The pair $(x^0, s^0)$ is given, such that $-Mx_0 + s_0 = q$.
We assume that $x_0 > 0,\ s_0 > 0$ as well as
$\delta(x^0, s^0, \mu^0) \le \tau,\ where\ \tau \in (0, 1)\ and\ \mu^0 = \frac{(x^0)^T s^0}{n}$.
$x = x^0, s = s^0$;
while $\|xs - w\| > \varepsilon$ do
    $\mu = (1 - \theta)\,\mu$;
    Determination of $(\Delta x, \Delta s)$ based on system (2)
    $x = x + \Delta x$;
    $s = s + \Delta s$;
end while
```

The input data of the algorithm are the starting interior points, which are denoted by the variables x^0 és s^0; the variable $0 < \theta < 1$, which is responsible for the reduction of the barrier parameter μ; the parameter $0 < \tau < 1$, which regulates the environment of the central path and the value $\epsilon > 0$, which determines the stopping criteria. In addition, it is necessary to enter the matrix M corresponding to the conditions of the problem, and the vector q. An initial value μ^0 must be assigned to the initial point pair (x^0, s^0). Starting from this pair, we can continue the cycle until the specified condition is met. Within the cycle, the value of μ must be reduced, and the simplest method for this is to multiply it by a number smaller than 1, in this case by $(1 - \theta)$. In addition, the directions Δx and Δs are calculated from system (2). In this case, we always define the new points x and s with a full-Newton step.

Since the implementation of full-Newton step for interior-point algorithms is usually not efficient enough, we make several modifications to obtain methods with lower running time. In the case of the algorithm we implemented, we use a multiplication factor $0 < \sigma \le 1$ for μ^0 to accelerate reduction. Furthermore, the current and maximum number of iterations and the parameters *ε, ρ*, and *σ* are also displayed as input data. To increase the efficiency of the implemented algorithm, we do not use a full-Newton step, but calculate the maximum step lengths *α(x)* and *α(s)* using the ratio test to the limit. Thereafter, we consider the minimum of *α(x)* and *α(s)* and the obtained positive real number *α* gives the length of the step, which is multiplied by a constant *ρ* between *0* and *1*.

Based on the current points *x* and *s*, *μ* is calculated as described in article [5], assuming that $e^T c \neq e^T w$.

Modified algorithm for WLCP

```
The vectors $x_0 > 0,\ s_0 > 0$ are given, such that
$-Mx_0 + s_0 = q$.
Let $0 < \sigma \le 1$ and $\mu^0 = \sigma \cdot \frac{(x^0)^T s^0}{n}$;
We assume that $\delta(x^0, s^0, \mu^0) \le \tau,\ where\ \tau \in (0,1)$.
Let $0 < \rho < 1$ and $\varepsilon > 0$.
$x = x^0,\quad s = s^0$;
$iter = 0,\quad max\_iter = 3000$;
do
    $\mu = \sigma \cdot \left| \frac{\mu_0 (x^T s - e^T w)}{e^T c - e^T w} \right|$;
    Determination of $(\Delta x, \Delta s)$ based on system (2);
    Determination of $\alpha(x)$ and $\alpha(s)$;
    $\alpha = \min\{\alpha(x), \alpha(s)\}$;
    $x = x + \rho \cdot \alpha \cdot \Delta x$;
    $s = s + \rho \cdot \alpha \cdot \Delta s$;
    $gap = \left| \frac{x^T s - e^T w}{e^T c - e^T w} \right|$;
    $infeasibility = \frac{\|Mx - s + q\|}{1 + \|q\|}$;
    $iter = iter + 1$;
while $((gap \ge \varepsilon\ or\ infeasibility \ge \varepsilon)\ and\ iter < max\_iter)$;
```

4. Numerical results

In the following, we present numerical results for various weighted complementarity problems.

1. Problem

The first problem is the same as problem 1 studied by Asadi and Mansouri [6], with the difference that instead of $w = 0$, we now consider a non-zero weight vector.

Let

$$n = 4,\ q = \begin{pmatrix} 4.0 \\ 4.0 \\ -2.0 \\ -1.0 \end{pmatrix},\ w = \begin{pmatrix} 15.0 \\ 10.0 \\ 12.0 \\ 23.0 \end{pmatrix},\ M = \begin{pmatrix} 1.0 & -2.0 & 1.0 & -1.0 \\ 2.0 & 0.0 & -2.0 & 1.0 \\ 1.0 & 2.0 & 0.0 & -3.0 \\ 2.0 & -1.0 & 3.0 & 3.0 \end{pmatrix}.$$

2. Problem

The second problem is the modified version of the problem 5.2 studied by Mansouri and Pirhaji [7]. In this case as well, we work with a weight vector $w \neq 0$:

$$\text{Let } n = 7,\ q = \begin{pmatrix} -1.0 \\ -3.0 \\ 1.0 \\ -1.0 \\ 5.0 \\ 4.0 \\ -1.5 \end{pmatrix},\ w = \begin{pmatrix} 10.0 \\ 12.0 \\ 15.0 \\ 35.0 \\ 15.5 \\ 12.0 \\ 20.2 \end{pmatrix},$$

$$M = \begin{pmatrix} 1.0 & 0.0 & -0.5 & 0.0 & 1.0 & 3.0 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.0 & 2.0 & 1.0 & -1.0 \\ -0.5 & 0.0 & 1.0 & 0.5 & 1.0 & 2.0 & -4.0 \\ 0.0 & 0.0 & 0.5 & 0.5 & 1.0 & -1.0 & 0.0 \\ -1.0 & -2.0 & -1.0 & -1.0 & 0.0 & 0.0 & 0.0 \\ -3.0 & -1.0 & -2.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 4.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix}.$$

3. Problem

The last problem is studied by Achache [8], but in this case we use the assumption $w \neq 0$ as well.

$$\text{Let } n = 500,\ q = \begin{pmatrix} -1.0 \\ -1.0 \\ \dots \\ \dots \\ -1.0 \\ -1.0 \end{pmatrix},\ w = \begin{pmatrix} v_0 \\ v_1 \\ \dots \\ \dots \\ v_{498} \\ v_{499} \end{pmatrix},$$

$$M = \begin{pmatrix} 1.0 & 2.0 & \dots & 2.0 & \dots & 2.0 \\ 0.0 & \ddots & 2.0 & \dots & 2.0 & \dots \\ \dots & 0.0 & 1.0 & 2.0 & \dots & 2.0 \\ 0.0 & \dots & 0.0 & 1.0 & 2.0 & \dots \\ \dots & 0.0 & \dots & 0.0 & \ddots & 2.0 \\ 0.0 & \dots & 0.0 & \dots & 0.0 & 1.0 \end{pmatrix},$$

where $v_i, 1 \le i \le n$ are randomly generated positive real numbers.

1. table. *Results for values of* $\varepsilon = 10^{-6},\ \rho = 0.95,\ \sigma = 0.3$.

Problem	Function φ	Size of matrix	Nr. of iterations	Running time
1	$\varphi(t) = t$	$n = 4$	6	1.382*ms*
1	$\varphi(t) = \sqrt{t}$	$n = 4$	14	1.947*ms*
2	$\varphi(t) = t$	$n = 7$	15	3.918*ms*

2	φ(t) = √t	$n = 7$	10	2.977ms
3	$\varphi(t) = t$	$n = 500$	19	7117.815ms
3	φ(t) = √t	$n = 500$	18	7474.826ms

To determine the directions, we first use the identical map function $\varphi(t) = t$, then the square root function $\varphi(t) = \sqrt{t}$. Furthermore, in the first case, let $\epsilon = 10^{-6}$, $\rho = 0.95$ and $\sigma = 0.3$, for which the obtained values are presented in table 1 in the case of WLCP problems.

2. table. *Results for values of* $\varepsilon = 10^{-8}$, $\rho = 0.95$, $\sigma = 0.3$.

Problem	Function φ	Size of matrix	Nr. of iterations	Running time
1	$\varphi(t) = t$	$n = 4$	10	1.614ms
1	φ(t) = √t	$n = 4$	19	2.362ms
2	$\varphi(t) = t$	$n = 7$	20	3.276ms
2	φ(t) = √t	$n = 7$	19	3.756ms
3	$\varphi(t) = t$	$n = 500$	23	9283.661ms
3	φ(t) = √t	$n = 500$	22	8764.180ms

It can be observed that in the case of the problem with a 4×4 matrix, when applying the square root function, we got the results in several iterations. The opposite can be said for the other two tasks (table 1).

Similar observations can be concluded for table 2 as for table 1, but by reducing the value of ε, the number of iterations increases, since during this modification we want to obtain the values of x and s with greater accuracy.

3. table. *Results for values of* $\varepsilon = 10^{-6}$, $\rho = 0.98$, $\sigma = 0.3$.

Problem	Function φ	Size of matrix	Iteration nr.	Running time
1	$\varphi(t) = t$	$n = 4$	11	1.646ms
1	φ(t) = √t	$n = 4$	14	2.134ms
2	$\varphi(t) = t$	$n = 7$	15	2.772ms
2	φ(t) = √t	$n = 7$	14	3.013ms
3	$\varphi(t) = t$	$n = 500$	18	7102.162ms
3	φ(t) = √t	$n = 500$	18	7119.939ms

In the case of increasing the value of ρ, the identical map function method proved to be the most effective for the first problem. The second problem was solved by the square root function method in fewer iterations, while in the case of the third problem, the number of iterations was the same (table 3).

4. table. *Results for values of* $\varepsilon = 10^{-6}$, $\rho = 0.95$, $\sigma = 0.9$.

Problem	Function φ	Size of matrix	Nr. of iterations	Running time
1	$\varphi(t) = t$	$n = 4$	138	15.872ms
1	φ(t) = √t	$n = 4$	140	21.080ms
2	$\varphi(t) = t$	$n = 7$	139	18.307ms
2	φ(t) = √t	$n = 7$	140	19.608ms
3	$\varphi(t) = t$	$n = 500$	143	100614.3ms
3	φ(t) = √t	$n = 500$	143	105317.7ms

Based on table 4, it can be concluded that in this case the program reached the solutions after several iterations, but this was expected, since by increasing the parameter σ we decelerated the decrease of the variable μ.

5. Conclusions

Regarding the solution of the WLCP, we introduced the algebraically equivalent transformation technique. With the help of an application written in Java programming language, we established that the function describing the direction can influence the results provided by the WLCP solver. The technique of algebraically equivalent transformation was examined from the perspective of the identical map and the square root function. It can be concluded that the number of iterations depends on the accuracy parameter ε, as well as on the values of the parameters σ, which reduces the barrier parameter, and ρ, which regulates the length of the step.

Acknowledgment

The authors express their gratitude to the Transylvanian Museum Association for the support provided to the research work.

References

[1] 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

[2] Asadi S., Darvay Zs., Lesaja G., Mahdavi-Amiri N.: *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

[3] Darvay Zs.: *A new algorithm for solving self-dual linear optimization problems*. Studia Universitatis Babeş-Bolyai, Series Informatica, 47/1 (2002) 15-26.

[4] Darvay Zs.: *New interior point algorithms in linear programming*, Advanced Modeling and Optimization, 5/1. (2003) 51–92.

[5] Darvay Zs., Orbán A.-Sz.: *Implementation of the full-Newton step algorithm for weighted linear complementarity problems*. Papers on Technical Science, 15/1. (2021) 15–18.
https://doi.org/10.33894/mtk-2021.15.04

[6] Asadi S., Mansouri H.: *Polynomial interior-point algorithm for $P_*(\kappa)$ horizontal linear complementarity problems*. Numerical Algorithms, 63/2. (2013) 385–398.
https://doi.org/10.1007/s11075-012-9628-0

[7] Mansouri H., Pirhaji M.: *A polynomial interior-point algorithm for monotone linear complementarity problems*. Journal of Optimization Theory and Applications, 157/2. (2013) 451–461.
https://doi.org/10.1007/s10957-012-0195-2

[8] Achache M.: *Complexity analysis and numerical implementation of a short-step primal-dual algorithm for linear complementarity problems*. Applied Mathematics and Computation, 216/7. (2010) 1889–1895.
https://doi.org/10.1016/j.amc.2010.03.015