Műszaki Tudományos Közlemények vol. 17. (2022) 11–15.
DOI: Hungarian: https://doi.org/10.33895/mtk-2022.17.03
English: https://doi.org/10.33894/mtk-2022.17.03

ALGEBRAICALLY EQUIVALENT TRANSFORMATION TECHNIQUE FOR WEIGHTED LINEAR COMPLEMENTARITY PROBLEMS

Zsolt DARVAY,[1] Zsanett JAKAB[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, 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 the case of the algebraically equivalent transformation, we divide the nonlinear equation of the system that characterizes 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 the 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 usual linear complementarity problem (*LCP*). *WLCP* was introduced by Potra [1] in 2012.

Let $\mathbb{R}^n_+$ be the set of n-dimensional vectors consisting of non-negative real numbers. Suppose that $M \in \mathbb{R}^{n\times n}$ is a monotone matrix, i.e. $x^T Mx \geq 0$ for each vector $x \in \mathbb{R}^n$.

Similar to the well-known 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 $x, s, q \in \mathbb{R}^n$ and $w \in \mathbb{R}^n_+$.

It can be observed that in the case of the weighted problem, the xs component-wise product 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:

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

where $\mu \in (0, \mu^0]$. The existence and uniqueness of the central path can be proved assuming that the matrix M is monotone and the starting interior point (x^0, s^0) exists [3].

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 all-one vector. Considering the algebraically equivalent transformation technique [4, 5], in the usual case, we proceed by dividing the centrality equation $xs = \mu e$ by the barrier parameter μ and then applying a continuously differentiable and invertible function $\varphi\colon (0, \infty) \to \mathbb{R}$ 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 by $xs = w(\mu)$ therefore we will divide it component-wise by the right-hand side vector $w(\mu)$. Moreover, 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, the search directions Δx and Δs can be uniquely determined by the system [3]:

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

It can be established that by introducing the function φ, the matrix of the system of linear equations that defines the search directions will always be the same, only the right-hand side vector depends on the function φ. In the implementation, we use two types of functions φ, 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 neighborhood of the given point on 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 theoretical and the implementation point of view.

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 obtaining 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

$\delta(x^0, s^0, \mu^0) \leq \tau$, 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 vectors x^0 and s^0; the variable $0 < \theta < 1$, which is responsible for the reduction of the barrier parameter μ; the parameter $0 < \tau < 1$, which adjusts the neighborhood of the central path and the value $\epsilon > 0$, which determines the stopping criteria. In addition, it is necessary to give the matrix M and the vector q corresponding to the conditions of the problem. 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 implemented version of the algorithm, we use a multiplication factor $0 < \sigma \leq 1$ for μ^0 to accelerate its 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 $\alpha(x)$ and $\alpha(s)$ using the minimum ratio test. Subsequently, we consider the minimum of $\alpha(x)$ and $\alpha(s)$ and the obtained positive real number α gives the length of the step, which is multiplied by a constant $\rho \in [0,1]$.

Based on the current points x and s, μ is calculated as described in article **[6],** 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 $0 < \sigma \leq 1$ and $\mu^0 = \dfrac{(x^0)^T s^0}{n}$.

We assume that $\delta(x^0, s^0, \mu^0) \leq \tau$, where $\tau \in (0,1)$.
Let $0 < \rho < 1$ and $\varepsilon > 0$.
$x = x^0$, $s = s^0$;
iter $= 0$, *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* $\geq \varepsilon$ *or infeasibility* $\geq \varepsilon$) *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 **[7]**, with the difference that instead of $w = 0$, we now consider a non-zero weight vector.

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

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

2. problem

The second problem is the modified version of problem 5.2 studied by Mansouri and Pirhaji **[8]**. In this case, we also work with a weight vector $w \neq 0$:

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

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

3. problem

The last problem is studied by Achache **[9]**, but in this case we use the assumption $w \neq 0$ as well.

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

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

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

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

It can be observed that in the case of the problem with a 4×4 matrix, when applying the square root function, we need more iterations than in case of the identical map. The opposite can be said for the other two problems (**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.

In the case of increasing the value of ρ, the method using the identical map proved to be the

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

Problem	Function φ	Size of matrix	Nr. of iterations	Running time (ms)
1	$\varphi(t) = t$	$n = 4$	6	1.382
1	$\varphi(t) = \sqrt{t}$	$n = 4$	14	1.947
2	$\varphi(t) = t$	$n = 7$	15	3.918
2	$\varphi(t) = \sqrt{t}$	$n = 7$	10	2.977
3	$\varphi(t) = t$	$n = 500$	19	7117.815
3	$\varphi(t) = \sqrt{t}$	$n = 500$	18	7474.826

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

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

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

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

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

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

most effective for the first problem. The second problem was solved by the method based on the square root function in fewer iterations, while in the case of the third problem, the number of iterations was the same (**Table 3.**).

Based on **Table 4**, it can be concluded that the program reached the solutions after more iterations than in the previous case, 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 the 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 analyzed 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 adjusts 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] Potra F. A.: *Sufficient Weighted Complementarity Problems*. Computational Optimization and Applications, 64/2. (2016) 467–488. https://doi.org/10.1007/s10589-015-9811-z

[4] Darvay Zs.: *A new Algorithm for Solving Self-Dual Linear Optimization Problems*. Studia Universitatis Babeş-Bolyai, Series Informatica, 47/1 (2002) 15–26.

[5] Darvay Zs.: *New Interior Point Algorithms in Linear Programming*. Advanced Modeling and Optimization, 5/1. (2003) 51–92.

[6] Darvay Zs., Orbán A.-Sz.: *Implementation of the Full-Newton Step Algorithm for Weighted Linear Complementarity Problems*. Műszaki Tudományos Közlemények, 15/1. (2021) 15–18. https://doi.org/10.33894/mtk-2021.15.04

[7] 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

[8] 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

[9] 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