

The s-Monotone Index Selection Rule for Criss-Cross Algorithms of Linear Complementarity Problems*

Zsolt Csizmadia[†] and Tibor Illés[‡] and Adrienn Nagy[§]

February 24, 2013

Abstract

In this paper we introduce the **s**-monotone index selection rules for the well-known criss-cross method for solving the linear complementarity problem (LCP). Most LCP solution methods require a priori information about the properties of the input matrix. One of the most general matrix properties often required for finiteness of the pivot algorithms (or polynomial complexity of interior point algorithms) is sufficiency. However, there is no known polynomial time method for checking the sufficiency of a matrix (classification of column sufficiency of a matrix is co-NP-complete).

Following the ideas of Fukuda, Namiki and Tamura, using Existentially Polynomial (EP)-type theorems, a simple extension of the criss-cross algorithm is introduced for LCPs with general matrices. Computational results obtained using the extended version of the criss-cross algorithm for bi-matrix games and for the Arrow-Debreu market equilibrium problem with different market size is presented.

2000 Mathematics Subject Classification: **49M35, 90C20.**

1 Introduction

Let us consider the linear complementarity problem (LCP) in the standard form: find vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, such that

$$-M\mathbf{u} + \mathbf{v} = \mathbf{q}, \quad \mathbf{u} \mathbf{v} = \mathbf{0}, \quad \mathbf{u}, \mathbf{v} \geq \mathbf{0} \quad (1)$$

where $M \in \mathbb{R}^{n \times n}$, $\mathbf{q} \in \mathbb{R}^n$ and $\mathbf{u} \mathbf{v} = (u_1 v_1, \dots, u_n v_n) \in \mathbb{R}^n$.

The linear complementarity problem is one of the most studied areas of mathematical programming. A large number of practical applications and the wide range of unsolved – both theoretical and algorithmic – problems make it an attractive field of research.

There are several different pivot algorithms to solve LCP problems with different matrices. The criss-cross algorithm is one of those which were developed independently – for different optimization problems – by Chang [4], Terlaky [23] and Wang [29]. Since then, the criss-cross method has become a class of algorithms that differs in the index selection rule.

*The research of Tibor Illés has been supported by the Hungarian National Research Fund OTKA No. T 049789.

[†]Xpress Development, FICO

[‡]Department of Differential Equations, Technical University Budapest

[§]Quality Assurance, Xpress, FICO

Akkeleş, Balogh and Illés [1] developed their criss-cross algorithm for LCP problems with bisymmetric matrices. They used the *LIFO* (last-in-first-out) and the *MOSV* (most-often-selected-variable) pivot rules. These index selection rules are flexible in the sense that they offer a choice of alternative pivots in certain situations, while still preserving finiteness. These index selection rules have been generalized for the linear programming problem in [9]. It is an interesting question which is the widest class of matrices for which the criss-cross algorithm with the above mentioned index selection rules can be extended, preserving finiteness.

The class of sufficient matrices was introduced by Cottle, Pang and Venkateswaran [6]. Sufficient matrices can be interpreted as generalizations of P and PSD matrices. Väliäho [27] showed that the class of sufficient matrices is the same as the class of P_* matrices, introduced in [20] for interior point methods of LCP problems. It was proved by den Hertog, Roos and Terlaky [14], that the sufficient matrices are exactly those matrices for which the criss-cross algorithm with the minimal index pivot selection rule solves the LCP problem with any right-hand side vector.

There is no known efficient algorithm to decide whether a matrix is sufficient or not, and Tseng [24] has shown that the classification of column sufficiency of a matrix is co-NP-complete. (Väliäho [26] developed a non-polynomial, inductive method to check sufficiency). Most algorithms developed for LCP problems have the practically unattractive property that they need the a priori information that the matrix is sufficient or possesses some other good properties. Fukuda, Namiki and Tamura [12] gave the first such algorithm – based on the alternative theorem for LCP of Fukuda and Terlaky [13], a generalization of the fundamental result of Farkas [10, 11] – used in the form of Existentially Polynomial (EP) theorems – that did not require a priori information on the sufficiency of the matrix. If the algorithm cannot proceed or would begin to cycle, it provides a polynomial size certificate that the input matrix is not sufficient.

For more than a decade, it was an open question whether EP theorems for linear complementarity problems could be proved using interior point algorithms or not. The first result in this direction has been published by Illés, M. Nagy and Terlaky [16]. Their EP theorem is different from that of Fukuda, Namiki and Tamura [12]: (i) there exist a solution for the dual LCP problem, (ii) there exist a solution for the feasibility problem part of the primal LCP, (iii) the matrix is not row sufficient. The polynomial size certificates for each of these statement is provided using a polynomial time interior point algorithm for linear feasibility problems. This result however is weaker than that of Fukuda, Namiki and Tamura since statement (ii) does not guarantee solvability of the primal LCP problem.

The following paper of Illés, M. Nagy and Terlaky [17] related to EP theorems contain a generalization of the affine scaling and predictor-corrector interior point algorithms to solve LCPs with general matrices in EP-sense, namely, the generalized interior point algorithms either solve the problem with rational a coefficient matrix in polynomial time or give a polynomial size certificate that the matrix does not belong to the set of $P_*(\kappa)$ matrices, with arbitrary large, but apriori fixed, rational, positive κ . This EP theorem differs both from the EP theorem published by Fukuda et al. [12] and from that of Illés et al. [17], because it contains only two statements. The reason for this is due to the fact that the embedding technique [22] of linear programming can not be generalized for LCPs with sufficient matrices. Therefore, one option is to deal with LCPs that have initial, strictly positive feasible solution. In this case, the dual LCP could not have solution, therefore the related EP theorems only need to consider two cases.

In their third paper, Illés, M. Nagy and Terlaky [18] use the embedding technique introduced

by Kojima et al. [20] and the related results about the solvability of the primal LCP to solve the embedded problem with a generalized path-following interior point algorithm. This allows them to prove almost the same EP theorem as Fukuda et al. [12]. There are two differences between the Fukuda-Namiki-Tamura and the Illés-Nagy-Terlaky EP theorems: (i) the later provides a decision between the options in polynomial time (ii) while, instead of the whole class of sufficient matrices, the algorithm works only for a given subset of sufficient matrices, i.e. for any $P_*(\kappa)$ matrix, with a priori fixed $\kappa > 0$. Property (i) is an improvement compared to the EP theorem of Fukuda et al., however (ii) states that the improvement works only for a given subset of sufficient matrices instead of the whole class.

The criss-cross method using the LIFO or MOSV index selection rules have been generalized for sufficient matrices to the form of EP-theorems in [8].

This paper introduces a new variant of the criss-cross type algorithm that combines the flexibility of the \mathbf{s} -monotone index selection rules with the practicability of EP-theorems: the criss-cross algorithm using \mathbf{s} -monotone index selection rules is modified in such a way that in case of an arbitrary matrix M and right-hand side \mathbf{q} , it either solves the LCP problem, or provides a polynomial size certificate that the matrix M is not sufficient. This property improves the value of the algorithm significantly making it applicable to a wide range of LCP problems, without requiring a priori information on the properties of the matrix. Indeed, the improved freedom of the \mathbf{s} -monotone pivot position selection compared to the traditional minimal index rule gives the possibility to avoid numerically instable pivots, further extending the practical applicability.

The finiteness proofs using \mathbf{s} -monotone index selection rules also proves the finiteness of these rules for the criss-cross method for the linear and convex quadratic problems in a more general context, as the Karush-Khun-Tucker conditions of the linear and quadratic problems yield a linear complementarity problem with bisymmetric matrices which are known to be sufficient (when the quadratic problem is convex) making this paper a natural extension of [9].

Throughout this paper, matrices are denoted by italic capital letters, vectors by bold, scalars by normal letters and index sets by capital calligraphic letters. Columns of a matrix are indexed as a subscript while rows are indexed by superscripts. M denotes the original problem matrix, \mathbf{q} the right hand side. Without loss of generality we may assume that $\text{rank}(M) = n$. The (ordered) set of all indices is $\mathcal{I} := \{1, 2, \dots, n, \bar{1}, \bar{2}, \dots, \bar{n}\} \cup \{q\}$ which includes the index of the right hand side, and so $|\mathcal{I}| = 2n + 1$. To denote the complementarity pairs, let $\bar{\alpha} = \alpha$ for all $\alpha \in \mathcal{I} \setminus \{q\}$, so the complementary index pair of $\bar{\alpha}$ is α . For a given basis B , we denote the nonbasic part of M by N ; the corresponding set of indices for the basis and nonbasic part are denoted by \mathcal{I}_B and \mathcal{I}_N , respectively. The corresponding short pivot tableau for B is denoted by $T := B^{-1}N$, while the transformed right hand side is denoted by $\bar{\mathbf{q}} := B^{-1}\mathbf{q}$. Individual coefficients of the short pivot tableau will be denoted by \bar{m}_{ij} ; please note that the definition of coefficients t_{ij} will be formally provided in Section 4.1.1. The complementary variable pairs (u_i, v_i) will usually be referred to as variable pairs. When it is advantageous, the element wise Hadamard product will be emphasized by " \cdot ".

The structure of the paper is as follows: following the introduction, the first two sections present the theoretical background used by the algorithms: Section 2 introduces the concept of sufficient matrices and their relevant algorithmically important properties, while Section 3 summarizes the s -monotone index selection rules. Section 4 presents new variants of the generalized criss-cross algorithm for LCP using \mathbf{s} -monotone index selection rules, while Section

5 extends it to the EP-theorem case. Section 6 present computational experiments solving small scale bimatrix games and Arrow-Debreu market equilibrium problems. The paper is closed by a summary.

2 Linear complementarity problems and sufficient matrices

For most solution methods for LCPs, the solvability of the linear complementarity problems and the efficiency of algorithms depend on the properties of matrix M . For pivot methods, one of the most general classes of interest is the *sufficient matrices*, which are generalizations of positive semidefinite matrices.

Definition 2.1 [6] *The matrix $M \in \mathbb{R}^{n \times n}$ is called column sufficient if no vector $\mathbf{x} \in \mathbb{R}^n$ exists, for which*

$$\begin{cases} x_i (M\mathbf{x})_i \leq 0 & \text{for all indices } i \in \{1, \dots, n\} \\ x_j (M\mathbf{x})_j < 0 & \text{for at least one index } j \in \{1, \dots, n\} \end{cases} \quad (2)$$

and we call it row sufficient if its transpose is column sufficient. A matrix M is called sufficient if it is both column and row sufficient at the same time.

It can be shown that column sufficient matrices are exactly those, for which the solution set of linear complementarity problems is convex [6], in fact a polyhedron; den Hertog, Roos and Terlaky [14] proved that sufficient matrices are exactly those, for which the criss-cross algorithm with the minimal index pivot rule can solve linear complementarity problems for any right-hand side vector \mathbf{q} , in a finite number of iterations.

The algorithms presented in this paper will use the concept of *strictly sign reversing* and *strictly sign preserving* vectors:

Definition 2.2 [12] *We call a vector $\mathbf{x} \in \mathbb{R}^{2n}$ strictly sign reversing if*

$$\begin{aligned} x_i x_{\bar{i}} &\leq 0 && \text{for all indices } i = 1, \dots, n \\ x_i x_{\bar{i}} &< 0 && \text{for at least one index } i \in \{1, \dots, n\}. \end{aligned}$$

We call a vector $\mathbf{x} \in \mathbb{R}^{2n}$ strictly sign preserving if

$$\begin{aligned} x_i x_{\bar{i}} &\geq 0 && \text{for all indices } i = 1, \dots, n \\ x_i x_{\bar{i}} &> 0 && \text{for at least one index } i \in \{1, \dots, n\}. \end{aligned}$$

To simplify the definition of the next lemma, let us introduce the subspaces

$$V := \{(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^{2n} \mid -M\mathbf{u} + \mathbf{v} = \mathbf{0}\}$$

and

$$V^\perp := \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{2n} \mid \mathbf{x} + M^T \mathbf{y} = \mathbf{0}\},$$

where $\mathbf{u}, \mathbf{v}, \mathbf{x}$ and \mathbf{y} are all vectors of length n . V and V^\perp are orthogonal complementary subspaces [12] of \mathbb{R}^{2n} .

Lemma 2.1 [12] *A matrix $M \in \mathbb{R}^{n \times n}$ is sufficient if and only if no strictly sign reversing vector exists in V and no strictly sign preserving vector exists in V^\perp .*

A basis B of the linear system $-M\mathbf{u} + \mathbf{v} = \mathbf{q}$ is called complementary, if for each index $i \in \mathcal{I}$ exactly one of the columns corresponding to variables v_i and u_i is in the basis. A short pivot tableau [19] is called complementary, if the corresponding basis is complementary. The next lemma shows the sign structure of short complementary pivot tableaux of LCPs with sufficient matrices, which is the main property of these matrices that the algorithms presented later will rely on.

Lemma 2.2 [6] *Let M be a sufficient matrix, B a complementary basis and $\bar{M} = [\bar{m}_{ij} \mid i \in J_B, j \in J_N]$ the corresponding short pivot tableau. Then*

- (a) $\bar{m}_{i\bar{i}} \geq 0$ for all $i \in J_B$; furthermore
- (b) for all $i \in J_B$, if $\bar{m}_{i\bar{i}} = 0$ then $\bar{m}_{i\bar{j}} = \bar{m}_{j\bar{i}} = 0$ or $\bar{m}_{i\bar{j}} \cdot \bar{m}_{j\bar{i}} < 0$ for all $j \in J_B, j \neq i$.

The proof of the previous lemma is constructive, so if the given structure of the matrix is violated, we can easily obtain the certificate from tableau \bar{M} , that M is not sufficient. The coding size of this certificate is bounded by a polynomial of the input length of matrix M [12, 8]

By the permutation of $M \in \mathbb{R}^{n \times n}$, we mean the matrix $P^T M P$, where P is a permutation matrix.

Lemma 2.3 [14] *Let $M \in \mathbb{R}^{n \times n}$ be a row (column) sufficient matrix. Then*

1. any permutation of matrix M is row (column) sufficient,
2. the product DMD is row (column) sufficient, where $D \in \mathbb{R}_+^{n \times n}$ is a positive diagonal matrix,
3. every principal submatrix of M is row (column) sufficient.

A sufficient matrix \bar{M} is also sufficient after any number of arbitrary principal pivots. The class of sufficient matrices is closed under principal block pivot operations [26], and as a consequence their properties are preserved during the criss-cross type algorithms, as the exchange pivot operations carried out by the criss-cross algorithms are equivalent to a block pivot operation of size 2×2 .

The matrix of Example 2.1 will be used in all examples of the paper.

Example 2.1 *As an example of a non-sufficient matrix, consider*

$$M = \begin{pmatrix} 1 & -1 & -1 & 0 & -1 \\ -1 & 2 & 0 & 0 & -1 \\ 1 & -2 & -1 & -2 & 0 \\ -4 & -1 & -1 & 2 & 4 \\ -1 & 0 & 2 & 0 & 1 \end{pmatrix}$$

This matrix is neither column, nor row sufficient. In this example, the operator \cdot denotes the Hadamard, element wise product. Consider the column vector

$$\mathbf{x}^T = (0 \ 0 \ 1 \ 0 \ 0)$$

then

$$\begin{aligned} \mathbf{x} \cdot (M\mathbf{x}) &= \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} \cdot \left[\begin{pmatrix} 1 & -1 & -1 & 0 & -1 \\ -1 & 2 & 0 & 0 & -1 \\ 1 & -2 & -1 & -2 & 0 \\ -4 & -1 & -1 & 2 & 4 \\ -1 & 0 & 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} \right] = \\ &= \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 0 \\ -1 \\ -1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

a strictly sign reversing vector. Similarly, for row vector

$$\mathbf{y} = (0 \ 0 \ 1 \ 1 \ 0)$$

we have

$$(\mathbf{y}M) \cdot \mathbf{y} = (-3 \ -3 \ -2 \ 0 \ 4) \cdot (0 \ 0 \ 1 \ 1 \ 0) = (0 \ 0 \ -2 \ 0 \ 0)$$

making \mathbf{y} a proof that M is not sufficient according to Definition 2.1.

The decision problem, whether an arbitrary linear complementarity problem has a solution or not, is in NP, and not always in co-NP [5], although for the class of sufficient matrices it belongs to co-NP, and can be stated using the dual of the LCP (1) problem:

$$(\mathbf{x}, \mathbf{y}) \in V(M, \mathbf{q})^\perp := \left\{ (\mathbf{x}, \mathbf{y}) \mid \begin{array}{l} \mathbf{x} + M^T \mathbf{y} = \mathbf{0}, \quad \mathbf{q}^T \mathbf{y} = -1 \\ \mathbf{x} \mathbf{y} = \mathbf{0}, \quad \mathbf{x}, \mathbf{y} \geq \mathbf{0} \end{array} \right\} \quad (D - LCP)$$

Using the definition of the dual, the alternative theorem of LCP problems is as follows.

Theorem 1 [13] *For a sufficient matrix $M \in \mathbb{R}^{n \times n}$ and a vector $\mathbf{q} \in \mathbb{R}^n$, exactly one of the following statements holds:*

- (1) *the (P-LCP) problem has a feasible complementary solution (\mathbf{u}, \mathbf{v}) ,*
- (2) *the (D-LCP) problem has a feasible complementary solution (\mathbf{x}, \mathbf{y}) .*

As a consequence, it is well-characterized when the (P – LCP) problem has no feasible complementary solution if the matrix M is sufficient and rational, as a polynomial size certificate can be given, namely the solution of the problem (D – LCP).

The alternative theorem for LCP of Fukuda and Terlaky is an interesting generalization of Farkas-lemma [10, 11]. Due to the fact, that primal and dual LCPs are more complicated problems, to show that exactly one of the alternative statements holds it is necessary to have information on the matrix property. Therefore, Fukuda and Terlaky’s result, comparing to that of Farkas’ needs an extra assumption, that the matrix M is sufficient.

3 The s-monotone index selection rule

The concept of s-monotone index selection rules have been introduced in [9], presented here for the LCP case. We say a variable moves during a pivot, if it either leaves, or enters the basis.

Definition 3.1 (Possible pivot sequence) *A sequence of index pairs*

$$\mathcal{S} = \{\mathcal{S}_k = (i_k, o_k) : i_k, o_k \in \mathbb{N} \text{ for some consecutive } k \in \mathbb{N}\},$$

is called a possible pivot sequence, if

- (i) $2n = \max\{\max_{k \in \mathbb{N}} i_k, \max_{k \in \mathbb{N}} o_k\}$,
- (ii) *there exists a (P-LCP) in standard form with $2n$ variables and the $\text{rank}(M) = n$, and*
- (iii) *(possibly infinite) pivot sequence, where the moving variable pairs of (P-LCP) correspond to the index pairs of \mathcal{S} .*

The index pairs of a possible pivot sequence are thus only required to comply with the basic and nonbasic status. It is now easy to show that

Proposition 3.1 *If a possible pivot sequence is not finite then there exists a (sub)set of indices, \mathcal{I}^* , that occurs infinitely many times in \mathcal{S} . ■*

The concept of *pivot index preference* describes how to select pivot positions when the primary algorithm offers flexibility:

Definition 3.2 (Pivot index preference) *A sequence of vectors $\mathbf{s}_k \in \mathbb{N}^{2n}$ is called a pivot index preference of an index selection rule, if in iteration j , in case of ambiguity according to a pivot selection rule, the index selection rule selects an index with highest value in \mathbf{s}_j among the candidates.*

The concept of s-monotone index selection rule [9] aims to formalize a common monotonicity property of several index selection rules.

Definition 3.3 (s-monotone index selection rule) Let $n \in \mathbb{N}$ be given. An index selection rule is called **s-monotone**, if

1. there exists a pivot index preference $\mathbf{s}_k \in \mathbb{N}^{2n}$, for which
 - (a) the values in the vector \mathbf{s}_{j-1} after iteration j may only change for i_j and o_j , where i_j and o_j are the indices involved in the pivot operation,
 - (b) the values may not decrease.
2. For any infinite possible pivot sequence \mathcal{S} and for any iteration j there exists iteration $r \geq j$ such that
 - (a) the index with minimal value in \mathbf{s}_r among $\mathcal{I}^* \cap \mathcal{I}_{B_r}$ is unique (let it be l), where \mathcal{I}_{B_r} is the set of basic indices, in iteration r ,
 - (b) in iteration $t > r$ when index $l \in \mathcal{I}^*$ occurs again in \mathcal{S} for the first time, the indices of \mathcal{I}^* that occurred in \mathcal{S} strictly between S_r and S_t have a value in \mathbf{s}_t higher than the index l .

To apply the **s-monotone** index selection rules for linear complementarity problems, a small extension is necessary comparing to that one for linear programming [9].

Definition 3.4 An **s-monotone** pivot rule applied to an LCP problem is symmetric, if $s(i) = s(\bar{i})$ always holds for any $i \in \mathcal{I}$.

The criss-cross algorithms presented in this paper will apply the **s-monotone** rules in a symmetric way. As examples for **s-monotone** rules, below is the definition of an **s** vector for the minimal index, the last-in-first-out and most often selected rules.

- *Minimal index/Bland*: The variables with the smallest index is selected.
Initialize **s** consisting of constant entries

$$s_i = n - i, i \in \mathcal{I}$$

where n is the number of the problem columns in the problem.

- *Last-In-First-Out (LIFO)*: The most recently moved variable is selected.
Initialize **s** equal to $\mathbf{0}$. For a pivot at (k, l) in the r^{th} iteration update **s** as

$$s'_i = \begin{cases} r & \text{if } i \in \{k, l\} \\ s_i & \text{otherwise} \end{cases}$$

- *Most-Often-Selected-Variable (MOSV)*: Select the variable that has been selected the largest amount of times before.

Initialize **s** equal to $\mathbf{0}$. For a pivot at (k, l) in the r^{th} iteration update **s** as

$$s'_i = \begin{cases} s_i + 1 & \text{if } i \in \{k, l\} \\ s_i & \text{otherwise} \end{cases}$$

4 The criss-cross method

This section presents the criss-cross algorithm for linear complementarity problems using \mathbf{s} -monotone index selection rules.

The criss-cross algorithms for LCP problems need a starting complementary solution, for which $\mathbf{u} = \mathbf{0}$ and $\mathbf{v} = \mathbf{q}$ is a possible selection using the assumption that $\text{rank}(M) = n$. Starting from the initial complementarity solution, the criss-cross method performs a sequence of so called diagonal and exchange pivots.

If in any complementary basis during the algorithm v_j is a basic variable and the value of variable v_j is infeasible and $\bar{m}_{jj} < 0$, then the algorithm, may perform a *diagonal pivot* where variable u_j enters the basis while variable v_j leaves it.

If $\bar{m}_{jj} = 0$ and the algorithm select v_j to leave the bases, then it pivots on such an index k for which $\bar{m}_{jk} < 0$. The solution obtained after a pivot like this is not complementary any more, so to restore the complementarity of the solution the algorithm pivots on the position (k, j) as well. According to Lemma 2.2, in case of sufficient matrices $\bar{m}_{kj} > 0$ will hold in this situation. These two pivots together are called an *exchange pivot*.

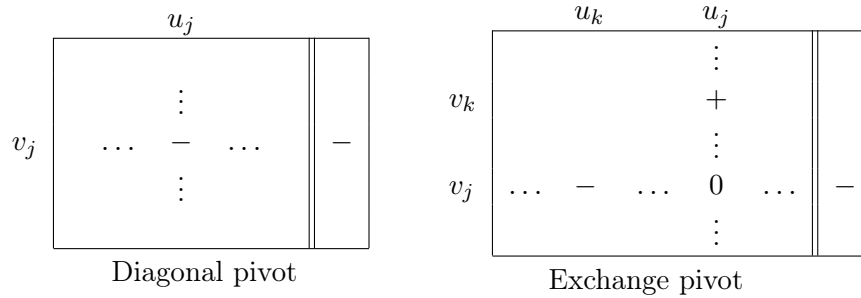


Figure 1: Diagonal and exchange pivot.

During an exchange pivot, variables u_j and u_k enter the basis, while v_j and v_k leave it. We say that u_j and v_j are chosen *actively*, while u_k and v_k are chosen *passively*. Thus, the terms passively and actively refer to the order in which the indices are selected, irrespective of the variable types (i.e. the same rule applies to an exchange pivot involving different combinations like (v_j, u_k) exchanged with (u_j, v_k)). The two types of pivot operations are presented in Figure 1.

When using \mathbf{s} -monotone index selection rules, the role of the \mathbf{s} -vector is to maintain a history about the movements of the variables. In the LCP case, these updates are generalized as follows: in case of an exchange pivot, the vector \mathbf{s} of the \mathbf{s} -monotone pivot rule is updated to the passively selected pair of variables first, and only then for the actively selected pair, thus the symmetry of vector \mathbf{s} (i.e. the value of any variable pair (u_i, v_i) in \mathbf{s} is the same after each pivot) is also maintained. This way, from the \mathbf{s} vector's point of view, an exchange pivot is considered as two pivots. This rule will play an important role in the finiteness proofs.

Note, that Lemma 2.3 ensures that the sufficiency of the matrix is preserved during the algorithm. Moreover, both the diagonal and the exchange pivot operations preserve complementarity as well.

The pseudo-code and flow chart of the criss-cross type algorithm using \mathbf{s} -monotone index selection rules is presented in Figure 2 and Figure 3.

Criss-cross type algorithm with symmetric \mathbf{s} -monotone pivot rule

Input: problem (1), where M is sufficient, $T := -M$, $\bar{q} := q$, $r := 1$. Initialize \mathbf{s} .
Begin
 $\mathcal{J} := \{i \in \mathcal{I} : \bar{q}_i < 0\}$.
While ($\mathcal{J} \neq \emptyset$) **do**
 $\mathcal{J}_{\max} := \{j \in \mathcal{J} \mid s(j) \geq s(\alpha), \text{ for all } \alpha \in \mathcal{J}\}$, let $k \in \mathcal{J}_{\max}$ arbitrary.
If ($\bar{m}_{kk} < 0$) **then**
Diagonal pivot on \bar{m}_{kk} , update vector \mathbf{s} for variable pair (u_k, v_k) .
Let $r := r + 1$.
Else
 $\mathcal{K} := \{i \in \mathcal{I} : \bar{m}_{ki} < 0\}$.
If ($\mathcal{K} = \emptyset$) **then** **Stop:** The LCP problem has no feasible solution.
Else
 $\mathcal{K}_{\max} = \{\beta \in \mathcal{K} \mid s(\beta) \geq s(\alpha), \text{ for all } \alpha \in \mathcal{K}\}$.
Let $l \in \mathcal{K}_{\max}$ arbitrary.
Exchange pivot on \bar{m}_{kl} and \bar{m}_{lk} .
Update vector \mathbf{s} for variable pair (u_l, v_l) as in iteration $r + 1$,
then for variable pair (u_k, v_k) as in iteration $r + 2$.
Let $r := r + 2$.
Endif
Endif
Endwhile
Stop: A feasible complementary solution has been computed.
End

Figure 2: The criss-cross type algorithm.

The sufficiency of matrix M ensures that (Lemma 2.2) the sign of the chosen pivot elements will be as desired in the case of exchange pivot. The algorithm terminates only if there is no solution or if it has found the solution, so it is sufficient to prove that it is *finite*. As the number of possible bases is finite, we have to show that the criss-cross type algorithm with \mathbf{s} -monotone pivot rule does not *cycle*.

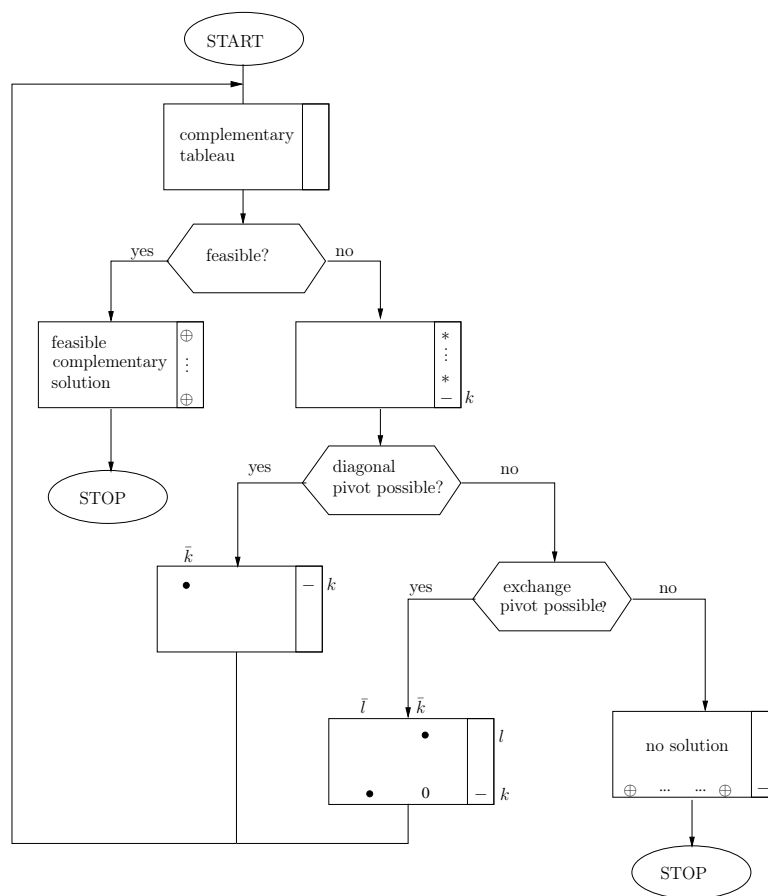


Figure 3: Flow chart of the algorithm.

4.1 Almost terminal tableaux of the criss-cross method

To prove finiteness of the algorithm, we assume the contrary, i.e. that it cycles. The results of this section are a generalization of the proof presented in [8] to the \mathbf{s} -monotone case.

Let us assume that an example exists for which the algorithm is not finite. The number of bases is finite, at most $\binom{2n}{n}$, so the algorithm makes an infinite number of iterations only if cycling occurs. Let us consider a minimal size cycling example. In such an example, because of minimality –and the monotonicity of \mathbf{s} – every variable moves during the cycle.

Let us consider the situation described by the second criterion of \mathbf{s} -monotonicity and with variable u_l outside the bases. Consider basis B' where variable u_l enters the bases for the first time after this situation occurred. Since u_l and v_l have the smallest value according to vector \mathbf{s} and the symmetry of the rule, v_l is the only infeasible variable in basis B' . The short pivot tableaux (presented in Figures 4 and 5) for this case are as follows:

1. The algorithm chooses u_l to enter the basis.

The diagonal element $\bar{m}_{ll} < 0$, so a diagonal pivot is possible [Figure 4, tableau (a)]: u_l enters the basis, while v_l leaves it. The vector \mathbf{s} is modified symmetrically for variable pair (u_l, v_l) .

(a)	<table style="border-collapse: collapse; margin: auto;"> <tr><td></td><td style="text-align: center;">u_l</td><td style="border-left: 1px solid black; padding-left: 5px;">\oplus</td></tr> <tr><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\oplus</td></tr> <tr><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\vdots</td></tr> <tr><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\vdots</td></tr> <tr><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\oplus</td></tr> <tr><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\oplus</td></tr> <tr><td style="padding-right: 5px;">v_l</td><td style="text-align: center;">$-$</td><td style="border-left: 1px solid black; padding-left: 5px;">$-$</td></tr> </table>		u_l	\oplus			\oplus			\vdots			\vdots			\oplus			\oplus	v_l	$-$	$-$
	u_l	\oplus																				
		\oplus																				
		\vdots																				
		\vdots																				
		\oplus																				
		\oplus																				
v_l	$-$	$-$																				

(b)	<table style="border-collapse: collapse; margin: auto;"> <tr><td></td><td style="text-align: center;">u_j</td><td style="text-align: center;">u_l</td><td style="border-left: 1px solid black; padding-left: 5px;">\oplus</td></tr> <tr><td style="padding-right: 5px;">v_j</td><td></td><td style="text-align: center;">$+$</td><td style="border-left: 1px solid black; padding-left: 5px;">\oplus</td></tr> <tr><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\vdots</td></tr> <tr><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\vdots</td></tr> <tr><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\oplus</td></tr> <tr><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\oplus</td></tr> <tr><td style="padding-right: 5px;">v_l</td><td style="text-align: center;">$-$</td><td style="text-align: center;">0</td><td style="border-left: 1px solid black; padding-left: 5px;">$-$</td></tr> </table>		u_j	u_l	\oplus	v_j		$+$	\oplus				\vdots				\vdots				\oplus				\oplus	v_l	$-$	0	$-$
	u_j	u_l	\oplus																										
v_j		$+$	\oplus																										
			\vdots																										
			\vdots																										
			\oplus																										
			\oplus																										
v_l	$-$	0	$-$																										

Figure 4: Variable u_l is actively selected to enter the bases.

2. The algorithm chooses variable u_l to enter the basis, but $\bar{m}_{ll} = 0$, so an exchange pivot is necessary [Figure 4, tableau (b)]. Variables u_l and u_j enter the basis, while variables v_l and v_j leave it. The vector \mathbf{s} is modified symmetrically (i.e. the complementary variable pairs always have the same value in \mathbf{s}), first for variable pair (u_j, v_j) and then for variable pair (u_l, v_l) as if in the next iteration.

The column of \mathbf{q} is the same as in tableau (a). In this case, it is not important whether u_j or v_j is in the basis. We consider the case when v_j is in the basis.

3. The algorithm chooses a variable u_j to enter the basis, but $\bar{m}_{jj} = 0$, so an exchange pivot is necessary and the algorithm chooses the variable u_l as well (passively), [Figure 5, tableau (c)].

(c)	<table style="border-collapse: collapse;"> <tr><td></td><td></td><td style="text-align: center;">u_l</td><td></td><td></td><td></td><td style="text-align: center;">u_j</td><td style="border-left: 1px solid black; padding-left: 5px;">\ominus</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\vdots</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\ominus</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">$+$</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\ominus</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\vdots</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\ominus</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\ominus</td></tr> <tr><td style="padding-right: 5px;">v_l</td><td></td><td></td><td></td><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\vdots</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\ominus</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td style="border-left: 1px solid black; padding-left: 5px;">\ominus</td></tr> <tr><td style="padding-right: 5px;">v_j</td><td style="text-align: center;">\oplus</td><td style="text-align: center;">\dots</td><td style="text-align: center;">\oplus</td><td style="text-align: center;">$-$</td><td style="text-align: center;">\oplus</td><td style="text-align: center;">\dots</td><td style="text-align: center;">\oplus</td><td style="border-left: 1px solid black; padding-left: 5px;">0</td><td style="border-left: 1px solid black; padding-left: 5px;">$-$</td></tr> </table>			u_l				u_j	\ominus								\vdots								\ominus								$+$								\ominus								\vdots								\ominus								\ominus	v_l							\vdots								\ominus								\ominus	v_j	\oplus	\dots	\oplus	$-$	\oplus	\dots	\oplus	0	$-$
		u_l				u_j	\ominus																																																																																												
							\vdots																																																																																												
							\ominus																																																																																												
							$+$																																																																																												
							\ominus																																																																																												
							\vdots																																																																																												
							\ominus																																																																																												
							\ominus																																																																																												
v_l							\vdots																																																																																												
							\ominus																																																																																												
							\ominus																																																																																												
v_j	\oplus	\dots	\oplus	$-$	\oplus	\dots	\oplus	0	$-$																																																																																										

Figure 5: Variable u_l is passively selected in a second pivot position in an exchange pivot to enter the basis.

According to the situation when u_l enters the basis, the row of v_j only in the columns of u_l and \mathbf{q} may contain negative elements, and because $\bar{m}_{jj} = 0$ –using Lemma 2.2– we also know the sign structure of the column of u_j . In this case, it is once again not important

whether u_j or v_j is in the basis. We consider the case when v_j is in the basis. The vector s is modified symmetrically first for variable pair (u_l, v_l) and then for variable pair (u_j, v_j) as if in the next iteration.

For the purpose of the EP-theorem considered in the next section, we note that in cases 1. and 2. only the properties of the pivot rule has been used when filling out the sign structures, while the sufficiency of the matrix has been used in the third case for the column of u_j .

Now consider basis B'' , when u_l leaves the basis for the first time after basis B' . The pivot tableau for this iteration can have three different structures (as presented in Figures 6 and 7), according to the second criterion of s -monotonicity.

- A.** According to the pivot rule, the algorithm chooses variable u_l to leave the basis, $\bar{m}_{ll} < 0$, so a diagonal pivot takes place [Figure 6, tableau (A)].

(A)	v_l \oplus \vdots \oplus $-$ \vdots $-$ $-$	(B)	v_k v_l u_k $+$ u_l $-$ 0 $-$
-----	--	-----	---

Figure 6: Variable u_l is actively selected to leave the bases.

- B.** The pivot rule chooses variable u_l to leave the basis, but $\bar{m}_{ll} = 0$, so an exchange pivot is needed: v_k (or u_k) enters the basis, while u_k (or v_k) leaves it [Figure 6, tableau (B)].
- C.** The algorithm chooses variable u_k (or v_k), but $\bar{m}_{kk} = 0$, so an exchange pivot takes place and v_l enters the basis, while u_k leaves it [Figure 7, tableau (C)].

(C)	v_l v_k u_l $+$ u_k $-$ 0 $-$
-----	---

Figure 7: Variable u_l is passively selected in a second pivot position in an exchange pivot to leave the basis.

We show that none of the tableaux (a) – (c) can be followed by one of the tableaux (A) – (C) if matrix M is sufficient. For the purposes of the next section that further extends the algorithm

for the EP-theorem case, it is important to make note which parts of the proofs are based on the sufficiency of the input matrix, and which on the properties of \mathbf{s} -monotonicity.

4.1.1 Auxiliary lemmas

We will use the following fundamental result of pivot tableaux called the orthogonality theorem that describes an orthogonal property between different pivot tableaux of the set of vectors. Denote the row vector

$$\left(\mathbf{t}^{(i)}\right)_k = \begin{cases} t_{ik}, & \text{if } k \in \mathcal{J}_N \cup \{b\} \\ 1, & \text{if } k = i \\ 0, & \text{otherwise} \end{cases}$$

and the columns vector

$$\left(\mathbf{t}_j\right)_k = \begin{cases} t_{kj}, & \text{if } k \in \mathcal{J}_B \\ -1, & \text{if } k = j \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 2 [19] *For any matrix $A \in \mathbb{R}^{m \times n}$ and with arbitrary bases B' and B'' , the vectors $\mathbf{t}^{(i)}$ and \mathbf{t}_j'' belonging to basic tableaux $T_{B'}$ and $T_{B''}$ respectively, are orthogonal.*

First, consider the cases that do not use the sufficiency of matrix M . We begin by showing that tableau (c) cannot be followed by tableaux (A) or (B).

Lemma 4.1 *Let us denote the tableau of case (c) by $T_{B'}$ and the tableau of A (or B) by $T_{B''}$. Consider the vectors $\mathbf{t}^{(\bar{x})}$ and \mathbf{t}_q'' , read from the row of the basic variable v_j in tableau $T_{B'}$, and from the column of \mathbf{q} in $T_{B''}$. Then*

$$\left(\mathbf{t}^{(\bar{j})}\right)^T \mathbf{t}_q'' > 0.$$

Proof. Let $\mathcal{K}'' := \{i \in \mathcal{I}_{B''} \mid \bar{q}_i'' < 0\}$. Using the second criterion of \mathbf{s} -monotone pivot rules, variables referring to these indices (with the possible exception of index \bar{j} and l) have not moved since B' , or else their \mathbf{s} value would have to be larger than of u_l , thus the algorithm would have chosen one from \mathcal{K}'' instead of u_l , so $\mathcal{K}'' \subseteq \mathcal{I}_{B'} \cap \mathcal{I}_{B''}$. This indicates that $t_{\bar{j}i}'' = 0$ for all indices $i \in \mathcal{K}'' \setminus \{\bar{j}, l\}$, thus

$$\sum_{i \in \mathcal{J}'' \setminus \{\bar{j}, l\}} t_{\bar{j}i}'' t_{iq}'' = 0. \quad (3)$$

Since at the exchange pivot made on B' , the \mathbf{s} value was first updated for variable pair (u_l, v_l) and only in the next pivot for (u_j, v_j) , we also know from \mathbf{s} -monotonicity that $t_{\bar{j}q}''$ and $t_{\bar{j}q}''$ are nonnegative (one is outside the basis, and since they have moved since B' , their \mathbf{s} value is bigger than of u_l).

Furthermore, it can be read from tableau (c) that $t_{\bar{j}j}'' = 0$, $t_{\bar{j}\bar{j}}'' = 1$, $t_{\bar{j}l}'' = 0$, $t_{\bar{j}l}'' < 0$ and $t_{\bar{j}q}'' < 0$ so

$$t_{\bar{j}\bar{j}}'' t_{\bar{j}q}'' + t_{\bar{j}j}'' t_{\bar{j}q}'' + t_{\bar{j}l}'' t_{lq}'' + t_{\bar{j}l}'' t_{lq}'' + t_{\bar{j}q}'' t_{qq}'' \geq t_{\bar{j}l}'' t_{lq}'' - t_{\bar{j}q}'' > 0, \quad (4)$$

because $t_{qq}'' = -1$ by definition, and $t_{lq}'' < 0$ according to the pivot rule of the algorithm (tableaux (A) and (B)).

If $h \notin \mathcal{K}'' \cup \{j, \bar{j}, l, \bar{l}, q\}$, we know again from the tableaux that $t'_{jh} \geq 0$ and by the definition of \mathcal{K}'' it holds that $t''_{hq} \geq 0$, so

$$\sum_{h \notin \mathcal{K}'' \cup \{j, \bar{j}, l, \bar{l}, q\}} t'_{jh} t''_{hq} \geq 0. \quad (5)$$

The result follows as we sum up inequalities (3)-(5). ■

From tableau (c) we considered the structure of the row for variable v_j , while from tableaux (A) and (B) the structure of the column of q . In none of these cases did we use the sufficiency of the matrix, and the proofs used only the combinatorial nature of the \mathbf{s} -monotone pivot rules. Thus tableaux (c) and (A) (or (B)) are exclusive because of the orthogonality theorem and the lemma above, regardless of the sufficiency of the matrix.

We now prove that tableaux (a) and (b) cannot be followed by tableau (C).

Lemma 4.2 *Let us denote tableau (a) (or (b)) by $T_{B'}$ and tableau (C) by $T_{B''}$. Consider the vectors \mathbf{t}'_q and $\mathbf{t}''^{(k)}$ belonging to the column of \mathbf{q} in tableau $M_{B'}$, and to the row of u_k in tableau $M_{B''}$. Then*

$$(\mathbf{t}''^{(k)})^T \mathbf{t}'_q > 0.$$

Proof. Like in the previous lemma, $\mathcal{K}''_k := \{i \in \mathcal{I}_{N''} \mid t''_{ki} < 0\} \subset \mathcal{I}_{N''}$ holds because of the second criterion of \mathbf{s} -monotonicity, so $t'_{iq} = 0$ for every $i \in \mathcal{K}''_k \setminus \{l\}$, thus

$$\sum_{i \in \mathcal{K}''_k \setminus \{l\}} t''_{ki} t'_{iq} = 0. \quad (6)$$

Furthermore, for an index $h \notin \mathcal{K}''_l \cup \{\bar{j}, j, \bar{l}, l, q\}$, $t''_{kh} \geq 0$ and $t'_{hq} \geq 0$, therefore

$$\sum_{j \notin \mathcal{K}''_l \cup \{\bar{j}, j, \bar{l}, l, q\}} t''_{kj} t'_{jq} \geq 0. \quad (7)$$

From tableaux $T_{B'}$ and $T_{B''}$ it can be read that

$t'_{qq} = -1$, $t''_{kk} = 1$, $t''_{k\bar{k}} = t''_{kl} = t'_{lq} = 0$ and $t''_{k\bar{l}} < 0$, $t''_{kq} < 0$, $t'_{lq} < 0$, $t'_{kq} \geq 0$ and $t'_{kq} \geq 0$ so

$$t''_{k\bar{k}} t'_{kq} + t''_{kk} t'_{kq} + t''_{k\bar{l}} t'_{lq} + t''_{kl} t'_{lq} + t''_{kq} t'_{qq} = t'_{kq} + t''_{k\bar{l}} t'_{lq} - t''_{kq} \geq t''_{k\bar{l}} t'_{lq} - t''_{kq} > 0. \quad (8)$$

The result follows as we sum up inequalities (6) – (8). ■

We can now consider tableaux where the sufficiency of the matrix plays an important role.

In the following, we show that tableaux (a) (or (b)) cannot be followed by tableaux (A) or (B).

Lemma 4.3 *Let the complementary solutions $(\mathbf{u}', \mathbf{v}')$ and $(\mathbf{u}'', \mathbf{v}'')$, belonging to tableaux (a) (or (b)) and (A) (or (B)) be given. Then the Hadamard product*

$$(\mathbf{u}' - \mathbf{u}'') \cdot M (\mathbf{u}' - \mathbf{u}'') \preceq \mathbf{0},$$

i.e. $(\mathbf{u}' - \mathbf{u}'')$ is a strictly sign reversing vector with respect to M .

Proof. We prove all four cases simultaneously.

$$\begin{aligned}
(\mathbf{u}' - \mathbf{u}'') \cdot M (\mathbf{u}' - \mathbf{u}'') &= (\mathbf{u}' - \mathbf{u}'') \cdot (\mathbf{q} + M \mathbf{u}' - \mathbf{q} - M \mathbf{u}'') \\
&= (\mathbf{u}' - \mathbf{u}'') \cdot (\mathbf{v}' - \mathbf{v}'') \\
&= \mathbf{u}' \cdot \mathbf{v}' - \mathbf{u}' \cdot \mathbf{v}'' - \mathbf{u}'' \cdot \mathbf{v}' + \mathbf{u}'' \cdot \mathbf{v}'' \\
&= -\mathbf{u}' \cdot \mathbf{v}'' - \mathbf{u}'' \cdot \mathbf{v}',
\end{aligned}$$

where the last equation holds because of the complementarity of the given solutions.

Let $\mathcal{K}'' := \{i \in \mathcal{I}_{B''} \mid \bar{q}_i'' < 0\}$. As before, according to \mathbf{s} -monotonicity, variables indexed by \mathcal{K}'' have not moved since bases B' , or else the algorithm would have chosen one from them, thus for all $i \in \mathcal{K}'' \setminus \{l\}$, the value of u_i' (or v_i'') and u_i'' (or v_i') is zero:

$$u_i' v_i'' + u_i'' v_i' = 0. \quad (9)$$

From tableau (a) (or (b)), and tableau (A) (or (B)), it can be read that $u_l' = 0, v_l' < 0$ and $u_l'' < 0, v_l'' = 0$ so,

$$u_l' v_l'' + u_l'' v_l' > 0. \quad (10)$$

Furthermore, for any $h \notin \mathcal{K}''$ it holds that $u_h', v_h', u_h'', v_h'' \geq 0$, thus

$$u_h' v_h'' + u_h'' v_h' \geq 0. \quad (11)$$

To summarize, the vector $(\mathbf{u}' - \mathbf{u}'')$ is such that $(\mathbf{u}' - \mathbf{u}'') \cdot M (\mathbf{u}' - \mathbf{u}'') \not\leq \mathbf{0}$. ■

Note, that the proof is constructive because the vector $\mathbf{u}' - \mathbf{u}''$ proving the lack of sufficiency of our matrix can easily be obtained from the bases B' and B'' .

In the last auxiliary lemma, we consider the case when tableau (c) would be followed by tableau (C).

Lemma 4.4 *Let us denote tableau (c) by $T_{B'}$, and tableau (C) by $T_{B''}$. Consider the vectors \mathbf{t}'_j and $\mathbf{t}''^{(k)}$ belonging to the column of u_j in tableau $T_{B'}$ and to the row of u_k in tableau $T_{B''}$. Then*

$$(\mathbf{t}''^{(k)})^T \mathbf{t}'_j < 0.$$

Proof. Let $\mathcal{K}_k'' = \{i \in \mathcal{I}_{N''} : t_{ki}'' < 0\} \setminus \{j\}$. Using the second criterion of \mathbf{s} -monotone pivot rules again, the variables of the indices \mathcal{K}_k'' have not moved since B' , so $(\mathcal{I}_{N''} \setminus \mathcal{K}_k'') \subset \mathcal{I}_{B'}$ and $\mathcal{K}_k'' \subset \mathcal{I}_{N'}$, thus $t'_{ij} = 0$ if $i \in \mathcal{K}_k''$. Based on these observations, we have

$$\sum_{i \in \mathcal{K}_k'' \cup \{q\}} t_{ki}'' t'_{ij} = 0. \quad (12)$$

Furthermore, if $h \notin \mathcal{K}_k'' \cup \{q, l, \bar{l}, j, \bar{j}, k, \bar{k}\}$ then $t'_{hj} \leq 0$ according to tableau (c). By the definition of \mathcal{K}_k'' , $t''_{kh} \geq 0$, so

$$\sum_{h \notin \mathcal{K}_k'' \cup \{q, l, \bar{l}, j, \bar{j}, k, \bar{k}\}} t''_{kh} t'_{hj} \leq 0, \quad (13)$$

From tableaux $T_{B'}$ and $T_{B''}$, taking the definition of vector \mathbf{t} into consideration, it follows that

$$t'_{lj} = t''_{k\bar{k}} = t'_{jj} = t'_{qj} = t''_{kl} = 0, \quad t''_{kk} = 1, \quad t'_{jj} = -1 \quad \text{and} \quad t'_{kj} \leq 0, \quad t''_{kl} < 0, \quad t'_{lj} > 0$$

so

$$t''_{kq} t'_{qj} + t''_{kl} t'_{lj} + t''_{k\bar{l}} t'_{\bar{l}j} + t''_{kj} t'_{jj} + t''_{k\bar{j}} t'_{\bar{j}j} + t''_{kk} t'_{kj} + t''_{k\bar{k}} t'_{\bar{k}j} < -t''_{kj}. \quad (14)$$

By the definition of the algorithm, at the exchange pivot in tableau c , the first variable pair for which the update of the \mathbf{s} vector is applied is (u_l, v_l) , and only after for (u_j, v_j) . Hence variable (u_j, v_j) is already considered to be moved since B' , thus by the second criterion of \mathbf{s} -monotone pivot rules their \mathbf{s} value is bigger than of index l . This is only possible if $t''_{kj} \geq 0$ either because out of bases, or because of its associated \mathbf{s} value.

The result follows as we sum up inequalities (12) – (14). ■

4.2 Finiteness of the criss-cross method

In this section, we prove the finiteness of the criss-cross algorithm.

Theorem 3 *The criss-cross type algorithm with \mathbf{s} -monotone pivot rule is finite for the linear complementarity problem with sufficient matrices.*

Proof. Let us assume the contrary, that the algorithm is not finite. Because a linear complementarity problem has finitely many different bases, the algorithm can have an infinite number of iterations only if it is cycling. Then we have a cycling example. Let us choose from the cycling examples one with a minimal size. Then, every variable moves during the cycle. Taking the auxiliary lemmas into consideration, after variable u_l enters the basis after basis B' , it cannot leave it again:

If it enters in case (a) or (b) and leaves the basis in case (A) or (B), Lemma 4.3. contradicts the sufficiency of matrix M .

If it enters in case (c) and leaves the basis in case (A) or (B), Lemma 4.1. contradicts the orthogonality theorem.

If it enters in case (c) and leaves the basis in case (C), Lemma 4.4. contradicts the orthogonality theorem.

If it enters in case (a) or (b) and leaves the basis in case (C), Lemma 4.2. contradicts the orthogonality theorem.

All possible cases lead to a contradiction, therefore the algorithm is finite. ■

Figure 8 shows the cases in which the sufficiency of matrix T has been used in the proof of finiteness of the criss-cross type algorithm.

	(a)	(b)	(c)
(A)	*	*	
(B)	*	*	
(C)			*

Figure 8: The cases when sufficiency of the pivot matrix is used.

5 EP theorems and the linear complementarity problem

This section generalizes the algorithm in the sense of EP theorems. As motivation, Example 5.1 demonstrates that the criss-cross algorithm may solve a LCP problem even if the matrix is not sufficient.

Example 5.1 *To demonstrate the criss-cross method, consider the linear complementarity problem with the matrix presented in Example 2.1 with the corresponding short pivot tableau where the identity matrix corresponding to \mathbf{v} serves as an initial complementary basis.*

Solving the problem with the criss-cross method, pivoting first on diagonal elements (u_1, v_1) and (u_2, v_2) :

	u_1	u_2	u_3	u_4	u_5	
v_1	-1	1	1	0	1	-1
v_2	1	-2	0	0	1	-2
v_3	-1	2	1	2	0	16
v_4	4	1	1	-2	-4	6
v_5	1	0	-2	0	-1	4

Tableau 1.

	v_1	u_2	u_3	u_4	u_5	
u_1	-1	-1	-1	0	-1	1
v_2	1	-1	1	0	2	-3
v_3	-1	1	0	2	-1	17
v_4	4	5	5	-2	0	2
v_5	1	1	-1	0	0	3

Tableau 2.

then (u_4, v_4) :

	v_1	v_2	u_3	u_4	u_5	
u_1	-2	-1	-2	0	-3	4
u_2	-1	-1	-1	0	-2	3
v_3	0	1	1	2	1	14
v_4	9	5	10	-2	10	-13
v_5	2	1	0	0	2	0

Tableau 3.

	v_1	u_2	u_3	v_4	u_5	
u_1	-2	-1	-2	0	-3	4
u_2	-1	-1	-1	0	-2	3
v_3	9	6	11	1	11	1
u_4	-4.5	-2.5	-5	-0.5	-5	6.5
v_5	2	1	0	0	2	0

Tableau 4.

arriving at the solution $u_1 = 4, u_2 = 3, v_3 = 1, u_4 = 6.5$ and all other variables zero. The method has found a feasible complementary solution for a problem with non-sufficient matrix.

Example 5.1 compensated for an obvious matrix coefficient making the matrix non-sufficient (diagonal entry for (u_3, v_3)) with a large right hand side value. As example 5.2 shows, this is not necessary.

Example 5.2 *Consider the same M matrix as in Example 5.1, but with a different right hand side.*

	u_1	u_2	u_3	u_4	u_5	
v_1	-1	1	1	0	1	-1
v_2	1	-2	0	0	1	2
v_3	-1	2	1	2	0	-1
v_4	4	1	1	-2	-4	5
v_5	1	0	-2	0	-1	1

Tableau 1.

	v_1	u_2	u_3	u_4	u_5	
u_1	-1	-1	-1	0	-1	1
v_2	1	-1	1	0	2	1
v_3	-1	1	0	2	-1	0
v_4	4	5	5	-2	0	1
v_5	1	1	-1	0	0	0

Tableau 2.

Pivoting on (u_1, v_1) arrives at a feasible complementary solution, even though the row of diagonal with the incorrect sign in respect to sufficiency started out to be infeasible.

An EP theorem is usually a collection of several alternative possible statements, from which one always holds, and if any statement holds, then a polynomial size (in the length of the input data) certificate for it must exist. It may also be viewed as a general framework for making provable and practically applicable theories.

The general form of an EP (Existentially Polynomial time) theorem is as follows [3]:

$$[\forall \mathbf{x} : F_1(\mathbf{x}) \text{ or } F_2(\mathbf{x}) \text{ or } \dots \text{ or } F_k(\mathbf{x})]$$

where $F_i(\mathbf{x})$ is a statement of the form

$$F_i(\mathbf{x}) = [\exists \mathbf{y}_i \text{ for which } \|\mathbf{y}_i\| \leq \|\mathbf{x}\|^{n_i} \text{ and } f_i(\mathbf{x}, \mathbf{y}_i)].$$

The extended algorithm makes use of the following two theorems.

Theorem 4 [12] *Let the matrix $M \in \mathbb{R}^{n \times n}$ be not sufficient. In this case, a certificate exists that M is not sufficient, the coding size of which is polynomially bounded by the input length of matrix M .*

Theorem 5 [12]. *For any matrix $M \in \mathbb{Q}^{n \times n}$ and vector $\mathbf{q} \in \mathbb{Q}^n$, at least one of the following statements holds:*

- (1) *problem (P-LCP) has a complementary, feasible solution (\mathbf{u}, \mathbf{v}) the encoding size of which is polynomially bounded by the input length of matrix M and vector \mathbf{q} .*
- (2) *problem (D-LCP) has a complementary, feasible solution (\mathbf{x}, \mathbf{y}) the encoding size of which is polynomially bounded by the input length of matrix M and vector \mathbf{q} .*
- (3) *matrix M is not sufficient, and there is a certificate the encoding size of which is polynomially bounded by the input length of matrix M .*

Note that cases (1) and (2) are exclusive, while case (3) can hold alone or together with either case (1) or (2). It is a naturally arising condition that the entries of the matrix should be rational numbers.

We modify the extended criss-cross algorithm so that it either solves problem (P-LCP) or its dual, or proves the lack of sufficiency of the input matrix, giving a polynomial size certificate.

Lemma 2.2 ensures that the pivot operations can always be done if our matrix is sufficient, and if it is not, it provides the required certificate that matrix M is not sufficient.

The criss-cross type algorithm with \mathbf{s} -monotone index selection rules in the form of EP-theorems

Input: $T = -M$, $\bar{\mathbf{q}} = \mathbf{q}$, $r = 1$, Initialize Q and \mathbf{s} .

Begin

While $((\mathcal{J} := \{i \in \mathcal{I} \mid \bar{\mathbf{q}}_i < 0\}) \neq \emptyset)$ **do**

$\mathcal{J}_{\max} := \{\beta \in \mathcal{J} \mid \mathbf{s}(\beta) \geq \mathbf{s}(\alpha), \text{ for all } \alpha \in \mathcal{J}\}$.

Let $k \in \mathcal{J}_{\max}$ be arbitrary.

Check $-\mathbf{u}' \cdot \mathbf{v}'' - \mathbf{u}'' \cdot \mathbf{v}'$ with the help of $Q(k)$.

If $(-\mathbf{u}' \cdot \mathbf{v}'' - \mathbf{u}'' \cdot \mathbf{v}' \leq 0)$ **then**

Stop: M is not sufficient, certificate: $\mathbf{u}' - \mathbf{u}''$.

Endif

If $(t_{kk} < 0)$ **then**

Diagonal pivot on t_{kk} , update \mathbf{s} .

$Q(k) = [J_B, \bar{\mathbf{t}}_q]$, $r := r + 1$.

ElseIf $(t_{kk} > 0)$

Stop: M is not sufficient, create certificate.

Else $/* t_{kk} = 0 */$

$K := \{\alpha \in I \mid \bar{t}_{k\alpha} < 0\}$

If $(K = \emptyset)$ **then**

Stop: DLCP solution.

Else

$\mathcal{K}_{\max} = \{\beta \in K \mid \mathbf{s}(\beta) \geq \mathbf{s}(\alpha), \text{ for all } \alpha \in K\}$.

Let $l \in \mathcal{K}_{\max}$ be arbitrary.

If $((\mathbf{t}_k, \mathbf{t}^k)$ or $(\mathbf{t}_l, \mathbf{t}^l)$ sign structure is violated) **then**

Stop: M is not sufficient, create certificate.

Endif

Exchange pivot on t_{kl} and t_{lk} , update \mathbf{s} first for (u_k, v_k) , then for (u_l, v_l) as in a next iteration.

$Q(k) = [J_B, \bar{\mathbf{t}}_q]$, $Q(l) = [\emptyset, \mathbf{0}]$, $r := r + 2$.

Endif

Endif

EndWhile

Stop: we have a complementary feasible solution.

End

Figure 9: The criss-cross type algorithm with \mathbf{s} -monotone index selection rules in the form of EP-theorems.

As Example 5.3 shows, in the lack of sufficiency the criss-cross method may cycle.

Example 5.3 Consider the same M matrix as in Example 5.1 and Example 5.2 but with a another different right hand side.

The criss-cross algorithm first pivots on (u_1, v_1) . In the second tableau, an exchange pivot is necessary, pivoting on (u_3, v_5) and then (u_5, v_3) leading to Tableau 4.

	u_1	u_2	u_3	u_4	u_5	
v_1	-1	1	1	0	1	-1
v_2	1	-2	0	0	1	2
v_3	-1	2	1	2	0	0
v_4	4	1	1	-2	-4	10
v_5	1	0	-2	0	-1	0

Tableau 1.

	v_1	u_2	u_3	u_4	u_5	
u_1	-1	-1	-1	0	-1	1
v_2	1	-1	1	0	2	1
v_3	-1	1	0	2	-1	1
v_4	4	5	5	-2	0	6
v_5	1	1	-1	0	0	-1

Tableau 2.

On Tableau 4, another exchange pivot takes place, (v_3, u_5) and then (v_5, u_3) resulting in Tableau 6..

	v_1	u_2	v_5	u_4	u_5	
u_1	-2	-2	-1	0	-1	2
v_2	2	0	1	0	2	0
v_3	-1	1	0	2	-1	1
v_4	9	10	5	-2	0	1
u_3	-1	-1	-1	0	0	1

Tableau 3.

	v_1	u_2	v_5	u_4	v_3	
u_1	-1	-3	-1	-2	-1	1
v_2	0	2	1	4	2	2
u_5	1	-1	0	-2	-1	-1
v_4	9	10	5	-2	0	1
u_3	-1	-1	-1	0	0	1

Tableau 4.

	v_1	u_2	v_5	u_4	u_5	
u_1	-2	-2	-1	0	-1	2
v_2	2	0	1	0	2	0
v_3	-1	1	0	2	-1	1
v_4	9	10	5	-2	0	1
u_3	-1	-1	-1	0	0	1

Tableau 5.

	v_1	u_2	u_3	u_4	u_5	
u_1	-1	-1	-1	0	-1	1
v_2	1	-1	1	0	2	1
v_3	-1	1	0	2	-1	1
v_4	4	5	5	-2	0	6
v_5	1	1	-1	0	0	-1

Tableau 6.

Tableau 6 coincides with Tableau 2: the pivot choices have been unique as the negative right hand side values were unique: the algorithm cycles.

We now need to analyse the proofs of finiteness of the original algorithm.

Avoiding cycling: Note, that minimality of the cycling example is not necessary in the proof of finiteness of the original algorithm. Every proof remains valid for any cycling example, since those variables that do not move during a cycle do not change their basis status, thus have a zero value in the orthogonality theorem.

Consider an arbitrary cycling example. Let the index set of the variables involved in the cycling be R , and consider an iteration, when cycling has already begun. Let the basis B' such that satisfies the second criterion of \mathbf{s} -monotonicity with variable x_l .

In this case the structure of tableaux $M_{B'}$ and $M_{B''}$ restricted to indices R and vector \mathbf{q} is exactly like in case (a) – (c) and (A) – (C). Between these two tableaux, a variable whose index

is not in R has not moved. Thus, in the product of the vectors analysed in Lemmas 4.1, 4.2 and 4.4, for the indices not in R and not q exactly one of the corresponding variables is in basis, so the contribution to the product for these indices is always zero. For the same reason, in the product of $-\mathbf{u}' \cdot \mathbf{v}'' - \mathbf{u}'' \cdot \mathbf{v}'$ in Lemma 4.3, the entries for the indices not in R are each zero. This proves that the proofs are valid for an arbitrary cycling example.

Handling the lack of sufficiency: Sufficiency has only been used in Lemmas 4.3 and 4.4. This latter one used the sign property of sufficient matrices, based on Lemma 2.2. Therefore, if the algorithm checks that the required sign property is fulfilled during every exchange pivot (cases (c) and (C) refer to such pivots), tableau (c) cannot be followed by tableau (C), because of the orthogonality theorem. If the required sign structure is violated, the certificate that matrix M is not sufficient is provided by the same lemma.

There remain the cases of tableaux (a)-(b) and (A)-(B). Lemma 4.3 handled this case. The proof of the lemma is based on the product

$$-\mathbf{u}' \cdot \mathbf{v}'' - \mathbf{u}'' \cdot \mathbf{v}' \quad (15)$$

referring to such subsequent tableaux $M_{B'}$ and $M_{B''}$, where the same variable moves during both pivot operations, and in both cases this variable was chosen actively (that is, not as the second variable of an exchange pivot). Note, that we do not need the whole tableau here, the only information we use is the column of \mathbf{q} (the actual complementary solution) and the set of indices in the basis. If vector (15) is strictly sign reversing, then as in the note after Lemma 4.3, the evidence that matrix M is not sufficient is the vector $\mathbf{u}' - \mathbf{u}''$.

Let us introduce a list $Q(p)$ ($p = 1, \dots, n$). Two vectors of dimension n belongs to every entry of this list. At the beginning,

$$Q(p) := \begin{bmatrix} [1, \dots, n] \\ [0, \dots, 0] \end{bmatrix} \quad p = 1, \dots, n.$$

When variable u_l or v_l leaves the basis during a diagonal pivot or such an exchange pivot where this variable is active (variable selected first), we modify the value of $Q(l)$ in such a way, that we write the *indices* of variables in basis to the first vector before the actual pivot operation, while we write the *values* of variables in basis before the pivot operation to the second vector:

$$Q(l) := \begin{bmatrix} [\text{indices of variables in basis}] \\ [\text{values of variables in basis}] \end{bmatrix}$$

If variable u_l or v_l enters the basis passively (as the second variable of an exchange pivot), we modify the value of $Q(l)$ as:

$$Q(l) := \begin{bmatrix} [1, \dots, n] \\ [0, \dots, 0] \end{bmatrix}$$

An operation $Q(j) = [\{I\}, \{\mathbf{h}\}]$ means that to the entry of j in the list Q , we write list I to the place of basic indices, while the values of the vector \mathbf{h} in the place of \mathbf{q} .

Before the algorithm performs a pivot operation, it checks if the actively selected variable that enters the basis was chosen actively previously or not when it left the basis. If yes, with the help of list Q , it checks vector (15) and only after this does it modify list Q . Because the complementary pairs of variables move together during the pivot operations, it is not necessary to provide space for both of them in list Q .

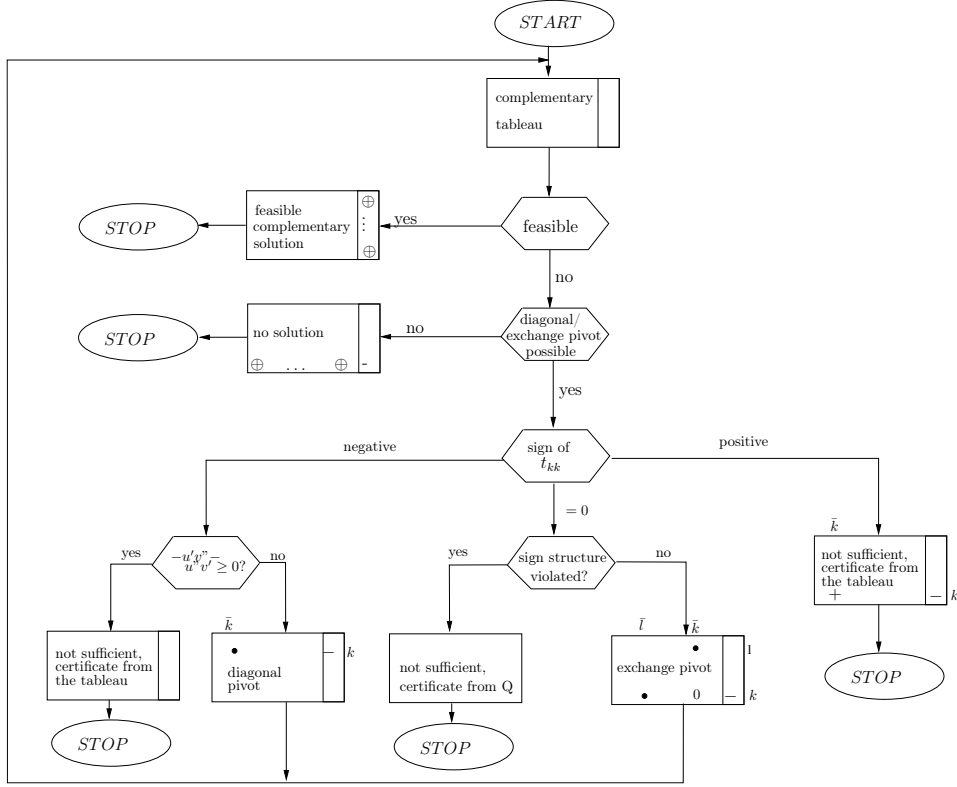


Figure 10: Flow chart of the modified criss-cross algorithm.

Note, that because the definition of the initial values of Q and the modification of Q during a passive exchange pivot, it suffices to check the product (15) during any pivot. If tableau (a) (or (b)) is followed by tableau (A) (or (B)), the product will always be zero.

It would not be necessary to fill out list Q every time. With a slight modification of the algorithm, we would be able to save storage space as well. In the worst case, the storage space required by list Q would be the storage space required to store n^2 integer and n^2 rational numbers.

We have to investigate the case when $(P - LCP)$ has no solution. This occurs when $K = \emptyset$. The structure of the pivot tableau is shown in Figure 11. Consider the vector

$$(\mathbf{x}', \mathbf{y}') = \mathbf{t}^{(k)} |_{J_N \cup J_B}.$$

Using the orthogonality theorem, we get that this vector is orthogonal to every row of $[-M^T | -I]$, in other words $M^T \mathbf{x}' + \mathbf{y}' = \mathbf{0}$. Applying the orthogonality theorem to the column of the right-hand side vector \mathbf{q} (in the starting basis), we have

$$(\mathbf{x}', \mathbf{y}')^T \mathbf{t}_q |_{J_N \cup J_B} = (\mathbf{x}', \mathbf{y}')^T (\mathbf{q}, \mathbf{0}) = \mathbf{x}'^T \mathbf{q} = q_k.$$

So the vector $(\mathbf{x}, \mathbf{y}) = (\mathbf{x}', \mathbf{y}')/(-q_k)$ is a solution to the problem $(D - LCP)$, because nonnegativity and complementarity follow from the structure of the pivot tableau.

Based on the discussion above, we showed that the modified criss-cross algorithm can be started from any complementarity basic solution of a general linear complementarity problem

without apriori information on the matrix property and using \mathbf{s} -monotone index rule, in a finite number of iteration the algorithm stops with one of the EP theorem cases, stated in Theorem 5.

Next example shows that the modified criss-cross algorithm (see on Figure 9 and it's flowchart on Figure 10) identifies the lack of sufficiency for matrix M.

	(\bar{u}, \bar{v})		\bar{q}
	$-M$	I	
$(\bar{x}, \bar{y}) \oplus$	$\dots \oplus 0 \oplus \dots \oplus$	1	$-$ k

$$(x, y)_i = \begin{cases} -t_{ki}/\bar{q}_k & \text{if } i \in I_N \\ -1/\bar{q}_k & \text{if } i = k \\ 0 & \text{otherwise.} \end{cases}$$

Figure 11: The dual solution when no primal solution exists.

Example 5.4 *As an example, consider the problem presented in Example 5.3 again. The extended algorithm would stop on Tableau 3: the sign of the second pivot position of the exchange pivot violates the expected sign structure.*

6 Computational experiences

In this section we provide some numerical experience using the proposed algorithm for solving general LCP problems arising from Arrow-Debreu exchange matrix problems and bimatrix games. The experiments have been carried out in Matlab, using the built in QR decomposition and update for the basis. To maximize the chance of success, in positions when the selected \mathbf{s} -monotone rule (MOSV) offered flexibility of pivot selection, a random position has been selected from among the eligible choices.

The results presented here emphasize the applicability of the new variant of the criss-cross method together with the flexibility of the \mathbf{s} -monotone index selection rules. For a comprehensive numerical study concentrating on the \mathbf{s} -monotone index selection rules see [15].

6.1 A market equilibrium problem

Consider the exchange market equilibrium problem as described by Walras [28]. There are m traders (players) and n goods on the market, where each good type j has a price $p_j \geq 0$. Each trader i is assumed to have an initial endowment of commodities

$\mathbf{w}_i = (w_{i1}, \dots, w_{in}) \in \mathbb{R}_{\oplus}^n$. The traders will sell their product on the market and use their income to buy a bundle of goods $\mathbf{x}_i = (x_{i1}, \dots, x_{in}) \in \mathbb{R}_{\oplus}^n$. A trader i has a utility function u_i , which describes his preferences for the different bundle of commodities and a budget constraint $\mathbf{p}^T \mathbf{x}_i \leq \mathbf{p}^T \mathbf{w}_i$. Finally, each trader i maximizes his individual utility function subject to his budget constraint.

Each trader optimizes his own utility function u_i with these side constraints:

$$\begin{aligned} \max \quad & u_i(\mathbf{x}_i) \\ \mathbf{p}^T \mathbf{x}_i \quad & \leq \quad \mathbf{p}^T \mathbf{w}_i \\ \mathbf{x}_i \quad & \geq \quad \mathbf{0}, \end{aligned}$$

where the vector of prices \mathbf{p} is an equilibrium for the exchange economy; if there is a bundle of goods $\mathbf{x}_i(\mathbf{p})$ (so a maximizer of the utility function u_i subject to the budget constraint) for all traders i , such that

$$\sum_{i=1}^m x_{ij}(\mathbf{p}) \leq \sum_{i=1}^m w_{ij} \quad \text{for all goods } j.$$

The exchange market equilibrium problem pursues prices where the demand $\sum_i x_{ij}(\mathbf{p})$ does not exceed the supply $\sum_i w_{ij}$ for any good j .

Arrow and Debreu [2] proved that under mild conditions, for concave utility functions, the exchange markets equilibrium exists. Using the Leontief utility function

$$u_i(\mathbf{x}_i) = \min_j \left\{ \frac{x_{ij}}{a_{ij}} : a_{ij} > 0 \right\},$$

where $A = (a_{ij}) \in \mathbb{R}_{\oplus}^{n \times n}$ is the Leontief coefficient matrix, Ye [25] has shown that the solution of the Arrow-Debreu competitive market equilibrium problem with Leontief's utility function is equivalent to the following linear complementarity problem:

$$A^T \mathbf{u} + \mathbf{v} = \mathbf{e}, \quad \mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0}, \quad \mathbf{u} \mathbf{v} = \mathbf{0} \quad \text{and} \quad \mathbf{u} \neq \mathbf{0}$$

where the matrix A has non-negative entries.

It is easy to see that this problem will almost always have a non-sufficient matrix, and as such is a demanding problem class for the generalized criss-cross algorithm.

The computational experiences have been carried out using a 100 problems, ranging $n \in \{10, 20, 40, 60, 80, 100, 200, 300, 400, 500\}$ taking 10 random instances for each value of n . For this problem the trivial basis corresponding to the columns of \mathbf{v} is not valid, as it is a feasible solution to the problem, but $\mathbf{u} \neq \mathbf{0}$ does not hold.

To address this problem, a structural crash heuristic has been used, based on the following heuristics. Start from the empty selection $B = \emptyset$. For any set of vectors, define its support as $\text{supp}(B) = \{j : \exists i : \mathbf{a}_i \in B, a_{j,i} \neq 0\}$. In each iteration of the crash heuristics for all $i \in 1 \dots n$, if there is such an index j for which $a_{ji} \neq 0$ and $i \notin \text{supp}(B)$, then add \mathbf{a}_i to B , else, add the corresponding identity vector e_i to the B . It is easy to see, that this procedure will yield a complementary basis, and unless $A = 0$ it will contain at least one columns corresponding to \mathbf{u} . When the selection of the column from A was not unique, the algorithm has selected randomly. Each experiment was repeated 10 times, yielding a total of 1000 test runs.

The number of successful solves diminished very quickly with size, also the algorithm terminates after a very small iteration count, with declaring the matrix not sufficient in most cases. There has been no successful solves for $n \geq 100$ suggesting that most of the successful solves for smaller sizes strongly depend on luck associated with the crash heuristics applied.

6.2 A bimatrix game

In this problem, two companies are considering entering n markets. Entering a market has both fixed, and variable costs, and there are fixed transport charges. The problem of finding optimal profit strategies can be formulated as a bimatrix game [21]. Consider a bimatrix game defined by A and B .

n	Successful solves	Mean iterations	Maximum iterations
10	762	0.619	12
20	318	1.432	12
40	26	1.445	9
60	13	1.479	10
80	2	1.396	9
100	0	1.295	10
200	0	1.213	9

Table 1: Numerical experiments on random market equilibrium problems.

Theorem 6 *The Karush-Khun-Tucker conditions of the bimatrix game can equivalently be formulated as*

$$\begin{aligned}
\mathbf{x}^T(A+B)\mathbf{y} - \alpha - \beta &\rightarrow \max \\
\mathbf{x}, \mathbf{y} &\geq \mathbf{0} \\
\mathbf{1}_m^T \mathbf{x} &= \mathbf{1} \\
\mathbf{1}_n^T \mathbf{y} &= \mathbf{1} \\
A\mathbf{y} &\leq \alpha \mathbf{1}_m \\
B^T \mathbf{x} &\leq \beta \mathbf{1}_n
\end{aligned}$$

where the zero valued solutions are the Nash-equilibria.

This Karush-Khun-Tucker conditions for this quadratic programming problem can be stated as

$$\begin{aligned}
\begin{pmatrix} Q & P^T \\ -P & O \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{t} \end{pmatrix} + \begin{pmatrix} \mathbf{s} \\ \mathbf{v} \end{pmatrix} &= \begin{pmatrix} \mathbf{c} \\ \mathbf{b} \end{pmatrix} \\
\mathbf{u} \mathbf{t} &= \mathbf{0} \\
\mathbf{s} \mathbf{v} &= \mathbf{0} \\
\mathbf{u}, \mathbf{t}, \mathbf{s}, \mathbf{v} &\geq \mathbf{0}
\end{aligned}$$

where

$$\begin{aligned}
Q &= \begin{pmatrix} O & -A-B \\ (-A-B)^T & O \end{pmatrix} & \mathbf{u} &= \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \in \mathbb{R}^{m+n} \\
P &= \begin{pmatrix} B^T & O & 0 & 0 & -1 & 1 \\ O & A & -1 & 1 & 0 & 0 \\ \mathbf{1}^T & \mathbf{0} & 0 & 0 & 0 & 0 \\ -\mathbf{1}^T & \mathbf{0} & 0 & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{1}^T & 0 & 0 & 0 & 0 \\ \mathbf{0} & -\mathbf{1}^T & 0 & 0 & 0 & 0 \end{pmatrix} & \mathbf{t} &= \begin{pmatrix} \alpha^+ \\ \alpha^- \\ \beta^+ \\ \beta^- \end{pmatrix} \in \mathbb{R}^4.
\end{aligned}$$

Payoff matrix size	Local solution (obj > 0.001)	Close to zero (obj < 0.001)	Local solution (obj > 0.01)	Close to zero (obj < 0.01)
2x2	563	437	516	484
3x3	768	232	684	316
4x4	908	92	801	199
5x5	931	69	827	173
6x6	960	40	855	145
7x7	973	27	871	129

Table 2: Numerical experiments on random bimatrix games.

As A and B can be chosen arbitrary, it is easy to see that the resulting LCP will not necessarily be sufficient.

Experiments have been carried out for payoff matrices with $n = m = 2, 3, \dots, 7$, with each experiment using random values for A and B and repeated 1000 times. As a trivial initial complementary bases, the identity columns corresponding to variables (\mathbf{s}, \mathbf{v}) have been selected.

Although the number of successful solves where the objective of the original quadratic problem is zero diminishes, the algorithm managed to find solution in a reasonable portion of the problems. Computational results are summarized on Table 2, where the optimal objective function value, due to numerical computational errors, claimed to be optimal in the interval $[0, \varepsilon)$. In case of column two of the Table 2, $\varepsilon = 0.001$, while in column three $\varepsilon = 0.01$. This shows that the computational precision may influence which problem will be declared as solved problem.

7 Summary

We have presented a variant of a generalized criss-cross type algorithm for linear complementarity problems with sufficient matrices, using \mathbf{s} -monotone pivot rules. For better practical applicability, we have modified the generalized algorithm so that the a priori information on the sufficiency of the matrix is not necessary. In case of lack of sufficiency, if the algorithm cannot ensure finiteness, then it terminates and provides a polynomial size certificate that the matrix is not sufficient. We have achieved our goals using the duality theorem of linear complementarity problems [13] and with its EP theorem form [12]. With the use of flexible \mathbf{s} -monotone pivot rules, the algorithm provides significant freedom in choosing the pivot position (usually during the first part of the algorithm), making it possible to avoid some numerically instable pivots.

Some supporting evidence for the applicability of the proposed algorithm has been presented by solving general linear complementarity problems arising from bimatrix games and the Arrow-Debreu market equilibrium problem, where the algorithm has proved to be applicable in (rather) small dimensions.

Acknowledgements

This research has been supported by the TÁMOP-4.2.2./B-10/1-2010-0009, Hungarian National Office of Research and Technology with the financial support of the European Union from the European Social Fund.

Tibor Illés acknowledges the research support obtained from Strathclyde University, Glasgow under the *John Anderson Research Leadership Program*.

Parts of this paper has been included in the Ph.D. thesis of Zsolt Csizmadia [7].

References

- [1] A.A. Akkeleş, L.Balogh, and T.Illés, New variants of the criss-cross method for linearly constrained convex quadratic programming, *European Journal of Operational Research*, **157:1** (2004), 74–86.
- [2] K. J. Arrow and G. Debreu, Existence of an equilibrium for competitive economy, *Econometrica*, **22** (1954), 265–290.
- [3] K. Cameron and J. Edmonds, Existentially polytime theorems, *In Polyhedral combinatorics* Morristown, NJ, 1989, *DIMACS Ser. Discrete Math. Theoretical Computer Science*, **1** (1990), 83–100.
- [4] Y.Y. Chang, Least index resolution of degeneracy in linear complementarity problems, *Technical Report 79-14*, Department of Operations Research, Stanford University, Stanford, California, USA, 1979.
- [5] S.J. Chung, NP-completeness of the linear complementarity problem, *Journal of Optimization Theory and Applications*, **60:3** (1989), 393–399.
- [6] R.W. Cottle, J.S. Pang, and V. Venkateswaran, Sufficient matrices and the linear complementarity problem, *Linear Algebra and its Applications*, **114/115** (1989), 231–249.
- [7] Z. Csizmadia, *New pivot based methods in linear optimization, and an application in petroleum industry*, www.cs.elte.hu/~csisza PhD Thesis, Eötvös Loránd University of Sciences, 2007.
- [8] Z. Csizmadia and T. Illés, New criss-cross type algorithms for linear complementarity problems with sufficient matrices, *Optimization Methods & Software*, **21:2** (2006), 247–266.
- [9] Z. Csizmadia, T. Illés, and A. Nagy, The s-monotone index selection rules for pivot algorithms of linear programming, *European Journal of Operation Research*, **221** (2012), 491–500.
- [10] Gy. Farkas, A fourier-féle mechanikai elv alkalmazásai (the applications of the mechanical principle of fourier), *Mathematikai és Természettudományi Értesítő*, **12** (1894), 457–472.
- [11] Gy. Farkas, Theorie der einfachen ungleichungen, *Journal für die Reine und Angewandte Mathematik*, **124** (1901), 1–27.
- [12] K. Fukuda, M. Namiki, and A. Tamura, EP theorems and linear complementarity problems, *Discrete Applied Mathematics. Combinatorial Algorithms, Optimization and Computer Science*, **84:1-3** (1998), 107–119.
- [13] K. Fukuda and T. Terlaky, Linear complementarity and oriented matroids, *Journal of the Operations Research Society of Japan*, **35:1** (1992), 45–61.

- [14] D. den Hertog, C. Roos, and T. Terlaky, The linear complementarity problem, sufficient matrices, and the criss-cross method, *Linear Algebra and its Applications*, **187** (1993), 1–14.
- [15] T. Illés, A. Nagy, Computational aspects of simplex and MBU-simplex algorithms using different anti-cycling pivot rules, *Department of Operations Research, Eötvös Loránd University of Sciences*, <http://mycite.omikk.bme.hu/doc/142879.pdf> *Operations Research Report*, **2012-02** (2012) submitted for publication.
- [16] T. Illés, M. Nagy and T. Terlaky, EP Theorem for Dual Linear Complementarity Problems, *Journal of Optimization Theory and Application*, **140** (2009) 233–238.
- [17] T. Illés, M. Nagy and T. Terlaky, Polynomial Interior Point Algorithms for General Linear Complementarity Problems, *Algorithmic Operations Research*, **5** (2010), 1–12.
- [18] T. Illés, M. Nagy and T. Terlaky, A polynomial path-following interior point algorithm for general linear complementarity problems, *Journal of Global Optimization*, **47** (2010), 329–342.
- [19] E. Klafszky and T. Terlaky, The role of pivoting in proving some fundamental theorems of linear algebra, *Linear Algebra and its Applications*, **151** (1991), 97–118.
- [20] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, A unified approach to interior point algorithms for linear complementarity problems, volume 538 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1991.
- [21] C.E. Lemke and J.T. Howson, Jr, Equilibrium points of bimatrix games, *Journal Soc. Indust. Appl. Math.*, **12** (1964), 413–423.
- [22] C. Roos, T. Terlaky, and J.-Ph. Vial, *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, USA, 1997. Second edition: *Interior Point Methods for Linear Optimization*, Springer, New York, 2006.
- [23] T. Terlaky, A convergent criss-cross method, *Optimization*, **16:5** (1985), 683–690.
- [24] P. Tseng, Co-NP-completeness of some matrix classification problems, *Mathematical Programming*, **Series A:88** (2000), 183–192.
- [25] Y. Ye, A Path to the Arrow-Debreu Competitive Market Equilibrium, *Math Programming*, **111:1-2** (2008), 315–348.
- [26] H. Väliäho, Criteria for sufficient matrices, *Linear Algebra and its Applications*, **233** (1996), 109–129.
- [27] H. Väliäho, P_* -matrices are just sufficient, *Linear Algebra and its Applications*, **239** (1996), 103–108.
- [28] L. Walras, *Elements of Pure Economics, or the Theory of Social Wealth*, (1874) (1899, 4th ed.; 1926, rev. ed., 1954, Engl. Transl.)
- [29] Z.M. Wang, A finite conformal-elimination free algorithm over oriented matroid programming, *Chinese Annals of Mathematics, Series B*, **8:1** (1987), 120–125.