

OPERATIONS RESEARCH
REPORT 2010-02

**The s-Monotone Index Selection Rules
for Pivot Algorithms of Linear
Programming**

Zsolt Csizmadia and Tibor Illés

May 2010

Eötvös Loránd University of Sciences
Department of Operations Research

**Copyright © 2010 Department of Operations Research,
Eötvös Loránd University of Sciences,
Budapest, Hungary**

<http://www.cs.elte.hu/opres/orr/>

ISSN 1215 - 5918

The s-Monotone Index Selection Rules for Pivot Algorithms of Linear Programming

Zsolt Csizmadia and Tibor Illés

Abstract

In this paper we introduce the concept of **s**-monotone index selection rule for linear programming problems. We show that several known anticycling pivot rules like the minimal index-, last-in-first-out- and the most-often-selected-variable pivot rules are **s**-monotone index selection rules. Furthermore, we show a possible way to define new **s**-monotone pivot rules. We prove that several known algorithms like the primal (dual) simplex- and MBU-simplex algorithms and criss-cross algorithm with **s**-monotone pivot rules are finite methods. Therefore, one possible research direction in the area of pivot algorithms might be to find **s**-monotone index selection rules that have interesting properties either from theoretical or from computational (for example larger flexibility in pivot selection) viewpoint.

Keywords: linear programming problem, pivot algorithms, anti-cycling pivot rules, **s**-monotone index selection rules.

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

1 Introduction

Let $A \in \mathbb{R}^{m \times n}$, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$ be matrix and vectors respectively, then

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

is a *primal linear programming (P-LP) problem*. While the *dual linear programming (D-LP) problem* can be defined as follows

$$\begin{aligned} \max \mathbf{b}^T \mathbf{y} \\ A^T \mathbf{y} \leq \mathbf{c} \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ are primal- and dual decision vectors, respectively.

Linear programming problems might be solved by applying pivot algorithms. The finiteness of the pivot algorithms depend on the algorithms property itself (pivot selection rule, anti-cycling strategy etc.).

Pivot based methods (like the simplex algorithm [6], MBU simplex algorithm [1] or the criss-cross algorithm [15, 11, 12]) often features the following similar principles:

1. The main flow of the algorithm is defined by a **pivot selection rule** which defines the basic characteristics of the algorithm, though the pivot position defined by it is not necessary unique (see for instance [6, 10, 4]), a series of "wrong" choices may even lead to cycling [10, 4].
2. To avoid the possibility of cycling, an **index selection rule** is used as an anti-cycling strategy (see for instance [3, 4, 13]), which may be flexible [5, 7] but usually at several basis during the algorithm, it defines the pivot position uniquely.

For several pivot algorithms – like simplex-, MBU simplex or criss-cross algorithms –, proofs of finiteness are often based on the orthogonality theorem [9, 2], considering a minimal cycling example [2], and following the movements of the least preferred variable of the index selection rule [3, 14, 7, 8]. Examples of such rules include

1. Pivot selection rules for (P-LP):
 - (a) **Simplex** [6] (Pivot column selection: negative reduced cost. Pivot element selection: using ratio test. Preserving non negativity of the right hand side.)
 - (b) **MBU simplex** [1] (Pivot column selection: negative reduced cost, choosing driving variable. Pivot element selection: defining driving and auxiliary pivots using primal and after that dual ratio tests. Monotone in the reduced cost of the driving variable.)

- (c) **Criss-cross** [12] (Pivot column/row selection is based on infeasibility – negative right hand side or negative reduced cost. Pivot element selection: admissible pivot positions.)

2. Index selection rules:

- (a) Bland's minimal index rule
 (b) last-in-first-out (LIFO)
 (c) most-often-selected-variable (MOSV)

LIFO and MOSV rules for linear programming problems were first used by S. Zhang [14] to prove the finiteness of the criss-cross algorithm with these anti-cycling index selection rules. Bilen, Csizmadia and Ildž's [2] proved that variants of MBU simplex algorithm are finite with both LIFO and MOSV index selection rules, while Csizmadia in his PhD Thesis [5] showed that the simplex algorithm is finite when the LIFO and MOSV are applied. These results led to the joint generalization of the above mentioned anti-cycling index selection rules.

Without loss of generality we may assume that the $\text{rank}(A) = m$. Let us associate to (P-LP) the (primal) pivot tableau

A	\mathbf{b}
\mathbf{c}^T	$*$

and let us assume that A_B is an $m \times m$ regular submatrix of the matrix A , thus form a basis of the linear system $A\mathbf{x} = \mathbf{b}$. In this case the (primal) basic pivot tableau associated with the (P-LP) problem and basis A_B is

$A_B^{-1}A$	$A_B^{-1}\mathbf{b}$
$\mathbf{c}^T - \mathbf{c}_B^T A_B^{-1}A$	$-\mathbf{c}_B^T A_B^{-1}\mathbf{b}$

The variables corresponding to the column vectors of the basis A_B are called *basic variables*. The index set of basic and nonbasic variables will be denoted by \mathcal{I}_B and \mathcal{I}_N , respectively. Let us introduce the following notations $T = A_B^{-1}A$, $\bar{\mathbf{b}} = A_B^{-1}\mathbf{b}$, $\bar{\mathbf{c}} = \mathbf{c} - \mathbf{c}_B^T A_B^{-1}A$. Now we are ready to define (column) vectors $\mathbf{t}^{(i)}$ and \mathbf{t}_j with dimension $(n+2)$, corresponding to the (primal) basic

tableau of the (P-LP) problem, where $i \in \mathcal{I}_B$ and $j \in \mathcal{I}_N$, respectively, in the following way:

$$(\mathbf{t}^{(i)})_k = t_{ik} = \begin{cases} t_{ik} & \text{if } k \in \mathcal{I}_B \cup \mathcal{I}_N \\ \bar{b}_i & \text{if } k = b \\ 0 & \text{if } k = c \end{cases}$$

and

$$(\mathbf{t}_j)_k = t_{kj} = \begin{cases} t_{kj} & \text{if } k \in \mathcal{I}_B \\ -1 & \text{if } k = j \\ 0 & \text{if } k \in (\mathcal{I}_N \setminus \{j\}) \cup \{b\} \\ \bar{c}_j & \text{if } k = c \end{cases}$$

where b and c denotes indices associated with vectors \mathbf{b} and \mathbf{c} , respectively. Furthermore, we define $\mathbf{t}^{(c)}$ and \mathbf{t}_b vectors in the following way

$$(\mathbf{t}^{(c)})_k = t_{ck} = \begin{cases} \bar{c}_k & \text{if } k \in \mathcal{I}_B \cup \mathcal{I}_N \\ 1 & \text{if } k = c \\ -\mathbf{c}_B^T A_B^{-1} \mathbf{b} & \text{if } k = b \end{cases}$$

and

$$(\mathbf{t}_b)_k = t_{kb} = \begin{cases} \bar{b}_k & \text{if } k \in \mathcal{I}_B \\ -1 & \text{if } k = b \\ 0 & \text{if } k \in \mathcal{I}_N \\ -\mathbf{c}_B^T A_B^{-1} \mathbf{b} & \text{if } k = c \end{cases}$$

and from now on we assume that c is always a basic index, while b is always a nonbasic index of the (P-LP) problem. Now we are ready to state the version of the orthogonality theorem that we frequently use in the finiteness proof of pivot algorithms that have anti-cycling index selection rules.

Result 1.1 (Orthogonality theorem, Klafszky and Terlaky, 1991). Let a (P-LP) problem be given, with $\text{rank}(A) = m$ and assume that $\mathcal{I}_{B'}$ and $\mathcal{I}_{B''}$ are two arbitrary bases of the problem. Then

$$(\mathbf{t}''^{(i)})^T \mathbf{t}'_j = 0$$

for all $i \in \mathcal{I}_{B''}$ and for all $j \notin \mathcal{I}_{B'}$.

The structure of this paper is organized in the following way: section 2 contains the necessary definitions that lead to the formal definition of the class of \mathbf{s} -monotone index selection rules. In section 3 we prove that some well-known index selection rules like the minimal index, LIFO and MOSV belongs

to the new wider class of index selection rules. In section 4 we prove that well-known pivot algorithms like the primal/dual simplex algorithm [6] and the primal/dual monotonic-build-up simplex algorithm [1] with *s*-monotone index selection rules are finite algorithms for solving linear programming problems. Without proof, we mention that the criss-cross algorithm [12] is also finite with *s*-monotone index selection rules. Some conclusions and further research questions close our paper.

2 The *s*-monotone index selection rule

In this section, we introduce a general framework for proving the finiteness of several pivot algorithms and index selection rule combinations mentioned in the previous section.

Definition 2.1 (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) $n = \max\{\max_{k \in \mathbb{N}} i_k, \max_{k \in \mathbb{N}} o_k\}$ is finite,
- (ii) there exists a $(P-LP)$ with n variables and the $\text{rank}(A) = m$, and
- (iii) (possibly infinite) pivot sequence, where the moving variable pairs of $(P - LP)$ correspond to the index pairs of \mathcal{S} .

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

Proposition 2.1. If a possible pivot sequence is not finite then there exists a (sub)set of indices, \mathcal{I}^* , that occur infinitely many times in \mathcal{S} . ■

Let us introduce the concept of *pivot index preference*.

Definition 2.2 (Pivot index preference). A sequence of vectors $\mathbf{s}_k \in \mathbb{N}^n$ 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 aims to formalize a common monotonicity property of several index selection rules.

Definition 2.3 (*s-monotone index selection rules*). Let $n \in \mathbb{N}$ be given. An index selection rule is called *s-monotone*, if

1. there exists a pivot index preference $\mathbf{s}_k \in \mathbb{N}^n$, 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 , and \mathcal{I}^* is the set of all indices that appear infinitely many times in \mathcal{S} ,
 - (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 .

3 s-monotone index selection rules

In this section we prove the following

Theorem 3.1. *The*

1. *minimal index rule,*
2. *the most-often-selected variable rule and*
3. *the last-in first-out index selection rule*

are s-monotone index selection rules.

The proof of this theorem follows from the following observation and two lemmas.

For the minimal index rule let us set each vector \mathbf{s}_k to be equal to the vector $(n, n-1, \dots, 1)^T$. Then it is easy to show that the minimal index rule is s-monotone.

Lemma 3.2. *The LIFO index selection rule is \mathbf{s} -monotone index selection rule.*

Proof. Let us initiate the vector \mathbf{s} to be the zero vector. In a pivot when x_{i_k} leaves and x_{o_k} enters the basis in the k^{th} iteration, the values of \mathbf{s} are modified to favor these variables:

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

and assume that a possible pivot sequence \mathcal{S} is generated using the pivot index preference.

It is clear that the series of \mathbf{s} vectors defined in such a way, form a pivot index preference for the LIFO rule. Furthermore, it is obvious that the properties 1 (a) and 1 (b) of the \mathbf{s} -monotone index selection rule are satisfied.

In case of an infinite possible pivot sequence and an arbitrary iteration j either all the s_i , $i \in \mathcal{I}^* \cap \mathcal{I}_{B_j}$ values are already different or if some have the same (initial) value, meaning they have not moved yet, then there should be an iteration later when these variables move for the first time. Let us denote that iteration by r when the last variable having index from \mathcal{I}^* moves for the first time. Then for the vector \mathbf{s}_r 2 (a) holds. Property 2 (b) follows from the definition of update for vector \mathbf{s} . \square

Now, we are ready to prove that the most-often-selected variable rule is an \mathbf{s} -monotone index selection rule, too.

Lemma 3.3. *The MOSV index selection rule is \mathbf{s} -monotone index selection rule.*

Proof. Let the vector \mathbf{s} be initialized as the zero vector. In a pivot when x_{i_k} leaves and x_{o_k} enters the basis in the k^{th} iteration, the values of \mathbf{s} are modified to increase the favor of these variables:

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

and assume that a possible pivot sequence \mathcal{S} is generated using the pivot index preference that defines the pivot index preference for MOSV. Due to the definition of the MOSV update, the properties 1 (a) and 1 (b) of the \mathbf{s} -monotone index selection rule is satisfied.

For any infinite possible pivot sequence, define \mathcal{I}^* as the set of indices appearing infinitely many times in the sequence. Let us denote by \mathcal{I}_{N_j} the

set of nonbasic indices for the j^{th} iteration and let $\mathcal{M}_N = \mathcal{I}_{N_j} \cap \mathcal{I}^*$ and $\mathcal{M}_B = \mathcal{I}^* \setminus \mathcal{M}_N$. We define the numbers γ_i as follows:

$$\gamma_i = \begin{cases} s_i, & \text{if } i \in \mathcal{M}_N \\ s_i + 1, & \text{if } i \in \mathcal{M}_B \end{cases}$$

Let

$$\mathcal{P} = \{i \in \mathcal{I}^* \mid i \in \arg \min_{k \in \mathcal{I}^*} \gamma_k\} \quad \text{and} \quad \min_{k \in \mathcal{I}^*} \gamma_k = \rho.$$

We continue to update \mathbf{s} according to the possible pivot sequence. Since $\mathcal{P} \subset \mathcal{I}^*$, thus for any $i \in \mathcal{P}$ there exists such an iteration, when variable x_i enters the basis for the first time after iteration j . When this happens, we delete its index from \mathcal{P} , thus $\mathcal{P} := \mathcal{P} \setminus \{i\}$. After finitely many iterations, such a set \mathcal{P} is obtained, for which $|\mathcal{P}| = |\{l\}| = 1$. After this happens, let the first iteration when variable x_l enters be r . We show that in iteration r the choice of x_l is unique. Observe, that in this case $s_l = \rho$, regardless whether x_l was moving in iteration j or not. Because of the pivot rule, $\rho < s_i$ if $i \in \mathcal{I}^* \setminus \mathcal{P}$ and since every variable with index $i \in \mathcal{P} \setminus \{l\}$ has at least once entered the basis after iteration j and now is outside the basis, their values in \mathbf{s} must be at least $\rho + 2$. On the other hand, if it was a basic variable in iteration j then its \mathbf{s} value is at least $\rho + 1$. Thus 2 (a) also holds.

Since the variable x_l enters the basis in iteration r , and every other variable with index in \mathcal{I}^* entering the basis after x_l already had a higher \mathbf{s} value than x_l in basis \mathcal{I}_{B_r} , according to the MOSV rule, thus 2 (b) also holds. \square

Analyzing the proofs of the previous two lemmas we can conclude that the 1 (a) and 1 (b) requirements of the definition of \mathbf{s} -monotone index selection rule are satisfied with the proper update strategy used to define the pivot index preference. Proving property 2 (a) there are three important ingredients: (i) the assumption that the pivot sequence is infinite, (ii) the finiteness of the index set, and (iii) the monotone increasing property of the pivot index preference, namely that for the vectors $\mathbf{s}_{k+1}, \mathbf{s}_k \in \mathbb{R}^n$: $\mathbf{s}_{k+1} \geq \mathbf{s}_k$ and $\mathbf{s}_{k+1} \neq \mathbf{s}_k$ hold for any iteration $k \in \mathbb{N}$. Property 2 (b) explains the changes in the \mathbf{s} -values of those variables that belong to the index set \mathcal{I}^* and have moved between two consecutive moves of the least preferred variable. This property depends strongly on the monotonicity of the pivot index preference and on the property 2 (a), too.

Now, we are ready to introduce generalizations of the MOSV and LIFO rules. Let us define these rules using their pivot index preferences.

Let the vector \mathbf{s} be initialized as the zero vector. In a pivot when x_{i_k} leaves and x_{o_k} enters the basis in the k^{th} iteration, the values of \mathbf{s} are modified to increase the favor of these variables.

Generalized-last-in-first-out rule (GLIFO): Let us consider a *strictly monotone increasing* sequence of positive rational numbers, namely for all $k \in \mathbb{N}$ indices $p_{k+1} > p_k$ hold.

$$s'_i = \begin{cases} p_k & \text{if } i \in \{i_k, o_k\}, \\ s_i & \text{otherwise,} \end{cases}$$

It is quite easy to show that this slight modification of the pivot index preference of LIFO rule will lead to an \mathbf{s} -monotone index selection rule, too. Namely, all the steps of the proof of Lemma 3.2, remains true; in fact after each variable that moves at all has moved at least once the sequences defined by GLIFO are the same as those by LIFO.

However, the generalization of MOSV define a significantly more general class of pivot sequences. We can generalize the MOSV rule as well by modifying its pivot index preference.

Generalized-most-often-selected-variable rule (GMOSV): Let us consider a *monotone increasing* sequence of positive rational numbers, namely for all $k \in \mathbb{N}$ indices $p_{k+1} \geq p_k$ hold.

$$s'_i = \begin{cases} s_i + p_k & \text{if } i \in \{i_k, o_k\}, \\ s_i & \text{otherwise,} \end{cases}$$

However, to show that GMOSV is an \mathbf{s} -monotone index selection rule we need a slightly more careful analysis of the proof of Lemma 3.3.

The requirements 1 (a) and 1 (b) are simply satisfied because of the definition of the corresponding pivot index preference. Justifying 2 (a), we need to modify the definition of γ_i introduced in the proof of Lemma 3.3, slightly. Let

$$\gamma_i = \begin{cases} s_i, & \text{if } i \in \mathcal{M}_N \\ s_i + p_j, & \text{if } i \in \mathcal{M}_B \end{cases}$$

since we would like to analyze the situation in the iteration j . Taking into consideration the monotone increasing nature of the p_k sequence we are able to identify - after finitely many iterations - the least preferred variable x_l in some iteration r .

Showing the uniqueness of the choice of x_l in the iteration r we need to do only a slightly more careful analysis of the situation. It remains true, that

$s_l = \rho$, regardless whether x_l was moving in iteration j or not. Furthermore, because of the monotone increasing nature of the p_k sequence, $\rho < s_i$ if $i \in \mathcal{I}^* \setminus \mathcal{P}$ and since every variable with index $i \in \mathcal{P} \setminus \{l\}$ has at least once entered the basis after iteration j and now is outside the basis, their values in \mathbf{s} must be at least $\rho + p_u + p_v$, where $r > u, v > j$. On the other hand, if it was a basic variable in iteration j than its \mathbf{s} value is at least $\rho + p_u$, where $r > u > j$. Since $p_u \geq p_j$ and $p_v \geq p_j$ hold we have verified that 2 (a) also holds.

Our last task is to show that 2 (b) holds as well. For this, let us collect all available information, namely we know that at the iteration r the following inequalities

$$s_l^r < s_i^r, \quad \forall i \in \mathcal{I}^* \setminus \{l\}$$

hold and that $s_l^t = s_l^r + p_r$. Let $i \in \mathcal{I}^* \setminus \{l\}$ be the index of such variable x_i that has moved between the iteration r and t , at least once, for instance in iteration k , where $t > k > r$ holds. Then

$$s_i^t \geq s_i^{k+1} = s_i^k + p_k \geq s_i^k + p_r \geq s_i^r + p_r > s_l^r + p_r = s_l^t,$$

thus 2 (b) is really satisfied.

Now we are ready to state our following result

Lemma 3.4. *GLIFO and GMOSV index selection rules are \mathbf{s} -monotone index selection rule.*

It is easy to check that when the sequence when $p_k = 1$, for all k , then GMOSV become MOSV, and that GMOSV is indeed a generalization of MOSV (e.g. it allows for switching from MOSV to LIFO).

4 Finiteness of pivot algorithms with \mathbf{s} -monotone index selection rules

In this section we show that simplex-, MBU simplex- and criss-cross algorithms with \mathbf{s} -monotone index selection rules are finite for linear programming problem. The proofs are based on the same ingredients: (i) assume contrary that the algorithm is cycling, namely that there are variables that enters/leaves bases infinitely many times, (ii) choose a minimal cycling example and denote the indices of those variables that moves infinitely many times by \mathcal{I}^* , (iii) apply the 2 (a) and 2 (b) properties of the \mathbf{s} -monotone index selection rule and identify the unique, least preferred variable, (iv) follow its moves, there are so-called almost terminal pivot tableaus corresponding to the

iterations when the least preferred variable enters/leaves bases, (v) during a cycle corresponding pairs of almost terminal tableaus should occur to complete the cycle, however this contradicts to the orthogonality theorem. Thus, in this framework, the only important step is to properly identify the almost terminal tableaus and to read out from the row/column the sign structure which will show the contradiction based on the orthogonality theorem. The structure of the almost terminal tableaus for different pivot algorithms might be slightly different.

4.1 The primal simplex algorithm with **s**-monotone index selection rules

We show the important part of the proof (i.e. defining the almost terminal tableaus and deriving the contradiction using the orthogonality theorem) for the primal simplex algorithm first, therefore let us present the pseudo-code of the simplex algorithm with **s**-monotone index selection rule.

The primal simplex algorithm with s-monotone index selection rules

Input: $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$ a feasible basis B , initialized \mathbf{s} vector.

Output: An optimal solution, or a certificate that the solution is unbounded.

Begin

$\mathcal{I}_N^- := \{i \in \mathcal{I}_N | \bar{c}_i < 0\}$.

While $\mathcal{I}_N^- \neq \emptyset$ **do**

Let $q \in \{i \in \mathcal{I}_N^- | \bar{c}_i < 0\}$ be arbitrary with maximal value respect to \mathbf{s} .

If $\mathbf{t}_q \leq 0$ **then**

Stop: problem is unbounded, certificate is \mathbf{t}_q .

Else

Let $\vartheta := \min \left\{ \frac{\bar{b}_i}{t_{iq}} \mid i \in \mathcal{I}_B, t_{iq} > 0 \right\}$ be the value of the primal ratio test.

Let $p \in \mathcal{I}_B$ arbitrary, such that $\frac{\bar{b}_p}{t_{pq}} = \vartheta$ and with maximal value respect to \mathbf{s} .

Endif

Pivot on (p, q) .

Endwhile

The solution is optimal.

End.

It is easy to verify that in a minimal cycling example all the variables are moving during a cycle and that the right hand side values are zeros (any nondegenerate pivot would improve the objective). Thus the minimal cycling example should be completely primal degenerate.

x_l				
+				0
				0
				\vdots
				0
				0
				\vdots
				0
-	\oplus	\dots	\oplus	

x_l enters the basis in basis B'

	x_k	
\mathcal{K}	+	0
	+	0
	\vdots	\vdots
	+	0
\mathcal{L}	\ominus	0
	\vdots	\vdots
	\ominus	0
	—	

x_l leaves the basis B''

Consider basis B' of the minimal cycling example, when the least preferred variable x_l enters the basis. According to the column selection rule of the simplex algorithm, the objective function row of the pivot tableau for basis B' has a negative entry for the nonbasic variable x_l and nonnegative entries for all other nonbasic variables. (This is the structure of our first almost terminal pivot tableau.)

In a minimal cycling example, since every variable moves infinitely many times, variable x_l must leave the basis. Consider basis B'' when x_l leaves the basis for the first time after B' . According to the s -monotone index selection rule, the choice of the leaving variable is selected from those basic variables that are least preferred, in our case x_l is a such variable. (This defines our second almost terminal pivot tableau.)

Consider the vector $\mathbf{t}'^{(c)}$ corresponding to the objective function row for basis B' and the vector \mathbf{t}''_k corresponding to the entering variable x_k for basis B'' . Let

$$\mathcal{K} = \{i \in \mathcal{I}_{B''} \mid t''_{ik} > 0\} \setminus \{l\}, \quad \text{and} \quad \mathcal{L} = \{j \in \mathcal{I}_{B''} \mid t''_{jk} \leq 0\}.$$

Then

$$\mathbf{t}^{(c)'}{}^T \mathbf{t}''_k = \sum_{i \in \mathcal{K}} t'_{ci} t''_{ik} + \sum_{j \in \mathcal{L}} t'_{cj} t''_{jk} + t'_{cl} t''_{lk} \leq t'_{cl} t''_{lk},$$

using that $t'_{cj} \geq 0$ and $t''_{jk} \leq 0$ for all $j \in \mathcal{L}$, and $t'_{ci} = 0$ for all $i \in \mathcal{K}$ because of the 1 (b) criterion, the values of \mathbf{s} may not decrease, and those variables that have moved since basis B' have a greater value in \mathbf{s} than variable x_l . By the 2 (b) criterion of s -monotone index selection rule, the variables corresponding to the index set \mathcal{K} , have not moved since basis B' , thus have a corresponding zero

value in $\mathbf{t}'^{(c)}$. Since $t'_{cl} < 0$ and $t''_{lk} > 0$, we have $\mathbf{t}^{(c)'}{}^T \mathbf{t}''_k < 0$, contradicting the orthogonality theorem. This proves that the primal simplex algorithm with \mathbf{s} -monotone index selection rules are finite.

4.2 The primal MBU simplex algorithm with \mathbf{s} -monotone index selection rules

The monotonic build-up simplex algorithm (MBU-SA) was introduced in [1]. It starts from a feasible basis and sets the feasibility of the dual variables one by one, while maintaining the feasibility of the already feasible dual variables. Although primal feasibility may be violated in some bases generated by the algorithm, primal feasibility is always restored when the selected driving variable becomes feasible.

The primal MBU simplex algorithm with s -monotone index selection rules

Input: $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$ a feasible basis B .

Output: An optimal solution, or a certificate that the solution is unbounded.

Begin

Initialize vector \mathbf{s} .

$\mathcal{I}_N^- := \{i \in \mathcal{I}_N | \bar{c}_i < 0\}$.

While $\mathcal{I}_N^- \neq \emptyset$ **do**

Let the driving variable $s \in \{i \in \mathcal{I}_N | \bar{c}_i < 0\}$ be arbitrary.

While $\bar{c}_s < 0$ **do**

Let $\mathcal{K}_s = \{i \in \mathcal{I}_B | t_{is} > 0\}$.

If $\mathcal{K}_s = \emptyset$ **then** Stop: problem is unbounded, certificate is \mathbf{t}_s .

Else

Let $\vartheta := \min \left\{ \frac{\bar{b}_i}{t_{is}} | i \in \mathcal{K}_s \right\}$ the value of the primal ratio test.

Let $r \in \mathcal{K}_s$ arbitrary such that $\frac{\bar{b}_r}{t_{rs}} = \vartheta$ and with maximal value respect to \mathbf{s} .

Let $\theta_1 := \frac{|\bar{c}_s|}{t_{rs}}$, and let $\mathcal{J} = \{i \in \mathcal{I}_N | \bar{c}_i \geq 0, t_{ri} < 0\}$.

If $\mathcal{J} = \emptyset$ **then**

$\theta_2 := \infty$.

Else

$\theta_2 := \min \left\{ \frac{\bar{c}_i}{|t_{ri}|} | i \in \mathcal{J} \right\}$ the value of the dual ratio test.

Let $q \in \mathcal{J}$ arbitrary such that $\theta_2 = \frac{\bar{c}_q}{|t_{rq}|}$ and with maximal value respect to \mathbf{s} .

Endif

If $\theta_1 \leq \theta_2$ **then**

Pivot on (r, s) , (*driving pivot*).

Else

Pivot on (r, q) , (*auxiliary pivot*).

Endif

Endif

Endwhile

Endwhile

The solution is optimal.

End

To establish the correctness of the algorithm, we repeat the key theorems proved in [1]. We will call a pivot in the column of the driving variable a *driving pivot*, while any other pivot an *auxiliary pivot*.

First, we state the theorem of Anstreicher and Terlaky [1] about the auxiliary pivots and their effects.

Result 4.1 (Anstreicher and Terlaky, 1994). Consider any pivot sequence produced by the primal MBU simplex algorithm corresponding to an initial feasible basis and the choice of a driving variable x_s . Then following an auxiliary pivot, the next basis produced by the algorithm has the following properties:

1. $\bar{c}_s < 0$,
2. if $\bar{b}_i < 0$, then $t_{is} < 0$,
3. $\max \left\{ \frac{\bar{b}_i}{t_{is}} \mid \bar{b}_i < 0 \right\} \leq \min \left\{ \frac{\bar{b}_i}{t_{is}} \mid t_{is} > 0 \right\}$.

It may happen that auxiliary pivot destroys primal feasibility. Next result of Anstreicher and Terlaky [1] states that the primal feasibility is restored after a driving pivot.

Result 4.2 (Anstreicher and Terlaky, 1994). Whenever the primal MBU simplex algorithm performs a driving pivot, the next basis is primal feasible.

Based on the previous theorems, we may state that the MBU algorithm is well-defined, [1].

Note that if the problem is both primal and dual nondegenerate, then finiteness is ensured by the fact that similarly to the simplex method, the objective function strictly increases in each iteration [1].

In the original paper [1], lexicography was used to ensure finiteness for degenerate problems. In this section, we prove that the algorithm is finite whenever s-monotone pivot rules are applied. First, we need to examine some further properties of the algorithm.

Lemma 4.1. *Both driving- and auxiliary pivots may only increase the reduced cost of the driving variable.*

Proof. A driving pivot makes the dual infeasible driving variable dual feasible, while an auxiliary pivot increases the reduced cost of the driving variable without making it nonnegative, or leaves it unchanged. (Follows from the Theorem 4.1.) \square

The next lemma states a further monotone property of the primal MBU simplex algorithm.

Lemma 4.2. *In any sequence of auxiliary pivots generated by the algorithm for driving variable x_r , the value $\max \left\{ \frac{\bar{b}_i}{t_{is}} \mid \bar{b}_i < 0 \right\}$ never decreases.*

Proof. Note that the third condition of Theorem 4.1 holds for any sequence of auxiliary pivots (see proof in [1]), thus

$$\max \left\{ \frac{\bar{b}_i}{t_{is}} \mid \bar{b}_i < 0 \right\} \leq \min \left\{ \frac{\bar{b}_i}{t_{is}} \mid t_{is} > 0 \right\} \quad (1)$$

always holds. Observe, that by the primal ratio test carried out by the algorithm, for an auxiliary pivot made on position (r, q) , the minimal ratio of $\min \left\{ \frac{\bar{b}_i}{t_{is}} \mid t_{is} > 0 \right\}$ is obtained, i.e.

$$\frac{\bar{b}_r}{t_{rq}} = \min \left\{ \frac{\bar{b}_i}{t_{is}} \mid t_{is} > 0 \right\}.$$

If we denote the tableau after the pivot on t_{rq} by \hat{T} and the new right hand side by $\hat{\mathbf{b}}$, then

$$\hat{t}_{rq} = \frac{t_{rq}}{t_{rq}} = 1, \quad \hat{b}_r = \frac{\bar{b}_r}{t_{rq}} < 0, \quad \frac{\hat{b}_r}{\hat{t}_{rq}} = \frac{\bar{b}_r}{t_{rq}}$$

since $t_{rq} < 0$ and $\bar{b}_r > 0$, thus since the auxiliary pivot is carried out on a negative pivot element, the new right-hand side value for index r becomes negative, while the ratio of the right-hand side and the pivot element remain the same. This means in the next iteration this ratio occurs in the left-hand side of (1). \square

Before examining a possible cycling example, we need a technical-type lemma. This lemma plays a fundamental role in the proof of finiteness with s -monotone pivot rules.

Lemma 4.3. *Let $a, b, \Theta \in \mathbb{R}$ such that $b \neq 0$ and $\frac{a}{b} = \Theta$. Let $c, d, \lambda \in \mathbb{R}$ such that $d \cdot \lambda \neq 0$ and $b + \lambda d \neq 0$. Then if $\frac{a+\lambda c}{b+\lambda d} = \Theta$, then $\frac{c}{d} = \Theta$.*

Proof.

$$\begin{aligned} \frac{a + \lambda c}{b + \lambda d} &= \Theta \\ a + \lambda c &= \Theta(b + \lambda d) \\ \Theta + \lambda \frac{c}{b} &= \Theta \left(1 + \lambda \frac{d}{b} \right) \\ \frac{c}{b} &= \Theta \frac{d}{b} \\ \frac{c}{d} &= \Theta. \end{aligned}$$

□

We are ready to prove that the MBU simplex algorithm with \mathbf{s} -monotone pivot rules is finite. Our proof is based on contradiction. Let us consider a minimal example, for which the algorithm is not finite. As usual, since the number of possible bases is finite, the algorithm must visit the same basis infinitely many times. It is clear, that because of minimality, in such an example each variable moves infinitely many times, with the possible exception of one single variable, which may remain an infeasible driving variable throughout the whole cycle.

Lemma 4.4. *Let us assume that we would like to solve a minimal cycling example using primal MBU simplex algorithm. The following properties hold:*

1. *Any basis generated by the algorithm is dual degenerate for all variables except one single variable. This variable remains the same throughout the algorithm and never enters the basis.*
2. *All variable moves infinitely many times, except one, which never enters the basis.*
3. *No driving pivot is made.*
4. *The primal ratio test always yields the same value. Furthermore, in any basis generated by the algorithm, for the column r of the driving variable, the ratio $\frac{\bar{b}_i}{t_{ir}}$ is the same for all i , where $t_{ir} \neq 0$.*

Proof. By Lemma 4.1, a pivot made in a nondegenerate column strictly increases the value of the driving variable. Observe, that while a pivot in a nondegenerate column leaves the column of the short pivot tableau nondegenerate, a degenerate pivot doesn't change the row of the objective function in the tableau.

It is easy to see, that in a cycling example, there exists an infeasible driving variable x_r that never becomes feasible, thus once this variable is selected for the role of a driving variable, only auxiliary pivots are made. Because the problem is a minimal cycling example, all other variables should move infinitely many times. However, by the observation made above, namely that $\bar{c}_i = 0$ for all $i \neq r$, so it follows that 1 holds.

Statements 2 and 3 follow immediately from 1.

Since the driving variable never changes, by Theorem 4.1 and Lemma 4.2, the value of the ratio test becomes a constant value after finitely many iterations. By the technical Lemma 4.3, it yields that the ratio must be the same for any basis generated by the algorithm. □

	x_r	x_l							
x_k	$+$	$-$	\oplus	\dots	\oplus				*
									\vdots
									\vdots
									*
									*
	$-$	0	0	\dots	0				\vdots
									*

	x_r	
$\mathcal{K} \left\{ \begin{array}{c} + \\ + \\ \vdots \\ + \end{array} \right.$	+	*
	+	*
	\vdots	\vdots
	+	*
$\mathcal{L} \left\{ \begin{array}{c} \ominus \\ \vdots \\ \ominus \end{array} \right.$	\ominus	*
	\vdots	\vdots
	\vdots	\vdots
	\ominus	*
	- 0 ... 0	

Figure 1: Almost terminal pivot tableaux for the MBU simplex algorithm.

Theorem 4.5. *The MBU simplex algorithm with \mathbf{s} -monotone index selection*

rule is finite.

Proof. Let us assume the contrary, and consider a minimal cycling example with entering variable x_l and leaving variable x_k in basis B' described in the second criterion of **s**-monotone index selection rules, and basis B'' when variable x_l leaves the basis for the first time after B' .

Consider vector $\mathbf{t}^{(k)}$ for basis B' and vector \mathbf{t}_r'' for basis B'' . Let

$$\mathcal{K} = \{i \in \mathcal{I}_{B''} \mid t_{ir}'' > 0\} \setminus \{l\}, \text{ and } \mathcal{L} = \{j \in \mathcal{I}_{B''} \mid t_{jr}'' \leq 0\}.$$

Then

$$\mathbf{t}^{(k)'}{}^T \mathbf{t}_r'' = \sum_{i \in \mathcal{K}} t_{ki}' t_{ir}'' + \sum_{j \in \mathcal{L}} t_{kj}' t_{jr}'' + t_{kr}' t_{rr}'' + t_{kl}' t_{lr}'' \leq t_{kr}' t_{rr}'' + t_{kl}' t_{lr}'',$$

using that $t_{kj}' \geq 0$ and $t_{jr}'' \leq 0$ for all $j \in \mathcal{L}$, and $t_{ki}' = 0$ for all $i \in \mathcal{K}$ because by the first criterion, the values of **s** may only increase, and those variables that have moved since B' have a greater value in **s** than variable x_l . By the third criterion of **s**-monotonicity, these variables have not moved since basis B' , thus have a corresponding zero value in $\mathbf{t}^{(c)}$. Since $t_{kl}' < 0$ and $t_{lr}'' > 0$, furthermore $t_{kr}' > 0$ and $t_{rr}'' = -1$, we have $\mathbf{t}^{(k)'}{}^T \mathbf{t}_r'' < 0$, contradicting the orthogonality theorem. \square

Similar arguments lead to the finiteness proofs of the dual simplex and dual MBU simplex algorithms with **s**-monotone index selection rule. The finiteness proof of the criss-cross algorithm with **s**-monotone index selection rule is only slightly different from the previous results. Important details of the finiteness proof for special **s**-monotone index selection rules can be found in [7, 11, 12]. Thus we can state the following quite general finiteness result of several pivot algorithms.

Theorem 4.6. *The primal (dual) simplex- and MBU simplex algorithms and the criss-cross algorithm with **s**-monotone index selection rules are finite for linear programming problems.*

In this way we unified several finiteness proofs [1, 2, 3, 7, 11, 12, 14] for the primal (dual) simplex- and MBU simplex- and criss-cross algorithms for linear programming problems. Furthermore, it is easy to show that the variants of MBU simplex- [2] and criss-cross [9] algorithms with **s**-monotone index selection rules are finite for the linear feasibility problems, too.

5 Conclusions and further research

Finiteness of the most known pivot algorithms depend on the anti-cycling pivot rules (see [4, 6, 10, 13]). In this paper we have shown that several known index selection rules (minimal index, LIFO, MOSV) possess same monotonicity type property, that has been captured by our new concept, the s -monotone index selection rules. Furthermore, we have introduced new, general anti-cycling pivot rules (GLIFO and GMOSV). We have unified the finiteness proof of some well-known pivot algorithms with s -monotone index selection rules.

Our new concept of s -monotone index selection rules and the related finiteness proofs show that anti-cycling pivot rules might leave some freedom of selecting the leaving/entering variable, especially at the initial phase of the computations. However, it is required even from the most flexible anti-cycling pivot rule, to build up an order among the variables at least in such a way that the selection of the least preferred variable become unique at some point of the computation. Form this follows that some strategies of selecting entering variable won't fulfill this requirement. However, we see some chances to compromise between two different goals: (i) decreasing the objective function in a greedy way, and (ii) guarantee the finiteness of the algorithm.

Suppose that we want to apply the *steepest edge* rule for selecting the entering variable and the *ratio test* for selecting the leaving variable in the (primal) simplex algorithm. Both in the selection of entering or/and leaving variable we might have multiple choices. It is known that the simplex algorithm with the steepest edge rule might cycling, due to primal degeneracy. Our suggestion for resolving such situation is the following: keep in mind that you want to have a finite pivot algorithm and apply the steepest edge rule if you have multiple choices for entering variables. Let us formalize a GLIFO and GMOSV index selection rule based on these ideas.

GLIFO with steepest edge index selection rule. Let us assume that $s_0 = \mathbf{0}$. In the k^{th} iteration let

$$\mathcal{C}_{k-1} = \{i \in \mathcal{I}_N : \bar{c}_i < 0\}, \quad \rho = \min_{i \in \mathcal{C}_{k-1}} s_{k-1,i} \quad \text{and} \quad \mathcal{S}_{k-1} = \{i \in \mathcal{C}_{k-1} : \rho = s_{k-1,i}\}.$$

If $|\mathcal{S}_{k-1}| = 1$ then we have no choice, the index of the entering variable is uniquely determined by finding the least preferred variable. However, until the order among variables - that reflects the previous iterations - is built, we might have $|\mathcal{S}_{k-1}| > 1$ and we can select the entering variable by using the

steepest edge index selection rule applied on \mathcal{S}_{k-1} only. Let

$$\gamma = - \min_{i \in \mathcal{S}_{k-1}} \frac{\bar{c}_i}{\|\mathbf{t}_i\|},$$

then we can define

$$p_k = \begin{cases} p_k + \delta & \text{if } p_{k-1} \geq \gamma, \\ \gamma & \text{if } p_{k-1} < \gamma, \end{cases}$$

where $\delta > 0$ is a given number. The sequence p_k satisfies the strictly monotonic increasing property, therefore we can use it to define, the new values of s_k as follows

$$s_{k,i} = \begin{cases} p_k & \text{if } i \in \{i_k, o_k\}, \\ s_i & \text{otherwise.} \end{cases}$$

Similarly we can define GMOSV with steepest edge index selection rule. The main difference is that in the definition of the sequence p_k , the number δ might be zero as well. Furthermore, the rule for updating the s_k values is exactly the same as in the general case for GMOSV.

In many implementations of the simplex algorithm, small random perturbations are used instead of index selection methods to ensure finiteness of the algorithm. One possible practical application of the ideas in this paper might be the simplex algorithms with arbitrary precision real number representations, where perturbation is impractical.

Acknowledgements. This research has been supported by the Hungarian National Research Fund OTKA No. T 049789. Research of Tibor Illés has been partially supported by the project "*Simulation and optimization: basic research in numerical mathematics*" sponsored by the frame TAMOP 4.2.2, 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*.

References

- [1] K. M. Anstreicher and T. Terlaky. A monotonic build-up simplex algorithm for linear programming. *Operations Research*, 42(3):556–561, 1994.

-
- [2] F. Bilen, Zs. Csizmadia, and T. Illés. Anstreicher-terlaky type monotonic simplex algorithms for linear feasibility problems. *Optimization Methods and Softwares*, 22(4):679–695, 2007.
 - [3] R.G. Bland. New finite pivoting rules for the simplex method. *Mathematics of Operations Research* 2:103-107, 1977.
 - [4] V. Chvátal. *Linear Programming*. W. H. Freeman and Company, New York, 1983.
 - [5] Zs. Csizmadia. *New pivot based methods in linear optimization, and an application in petroleum industry*. PhD Thesis, Eötvös Loránd University of Sciences, Budapest, Hungary, 2007. www.cs.elte.hu/~csisza
 - [6] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, New Jersey, 1963.
 - [7] T. Illés and K. Mészáros. A new and constructive proof of two basic results of linear programming. *Yugoslav Journal of Operations Research* 11(1):15-30, 2001.
 - [8] T. Illés and T. Terlaky. Pivot versus interior point methods: pros and cons. *European Journal of Operational Research*, 140(2):170-190, 2002.
 - [9] E. Klafszky and T. Terlaky. The role of pivoting in proving some fundamental theorems of linear algebra. *Linear Algebra and its Applications*, 151:97–118, 1991.
 - [10] K.G. Murty. *Linear and Combinatorial Programming*. John Wiley & Sons, Inc., New York, 1976.
 - [11] T. Terlaky. A finite “criss-cross method” for solving linear programming problems [in Hungarian]. *Alkalmazott Matematikai Lapok*, 10(3-4):289–296, 1984.
 - [12] T. Terlaky. A convergent criss-cross method. *Optimization*, 16(5):683–690, 1985.
 - [13] T. Terlaky and S. Zhang. Pivot rules for linear programming: a survey on recent theoretical developments. *Annals of Operations Research*, 46/47(1-4):203–233, 1993.
 - [14] S. Zhang. A new variant of criss-cross pivot algorithm for linear programming. *European Journal of Operational Research*, 116(3):607-614, 1997.
-

- [15] S. Zionts. The criss-cross method for solving linear programming problems. *Management Science*, 15(7):426–445, 1969.

Selected list of Operations Research Reports

- 2003-01** ZSOLT CSIZMADIA AND TIBOR ILLÉS, New criss-cross type algorithms for linear complementarity problems with sufficient matrices.
- 2003-02** TIBOR ILLÉS AND ÁDÁM B. NAGY, A sufficient optimality criteria for linearly constrained, separable concave minimization problems.
- 2004-01** TIBOR ILLÉS AND MARIANNA NAGY, The Mizuno–Todd–Ye predictor–corrector algorithm for sufficient matrix linear complementarity problem.
- 2005-03** BILEN FILIZ, ZSOLT CSIZMADIA AND TIBOR ILLÉS , Anstreicher–Terlaky type monotonic simplex algorithms for linear feasibility problems.
- 2005-04** TIBOR ILLÉS, MÁRTON MAKAI, ZSUZSANNA VAIK, Railway Engine Assignment Models Based on Combinatorial and Integer Programming.
- 2007-02** TIBOR ILLÉS, MARIANNA NAGY AND TAMÁS TERLAKY, An EP theorem for dual linear complementarity problem.
- 2007-03** TIBOR ILLÉS, MARIANNA NAGY AND TAMÁS TERLAKY, Polynomial interior point algorithms for general LCPs.