Criss-Cross Methods: A Fresh View on Pivot Algorithms

Komei Fukuda

Department of Mathematics, Swiss Federal Institute of Technology, Lausanne and Institute for Operations Research, Swiss Federal Institute of Technology, Zürich, Switzerland

Tamás Terlaky

Faculty of Technical Mathematics and Informatics, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands

Criss-cross methods are pivot algorithms that solve linear programming problems in one phase starting with any basic solution. The first finite criss-cross method was invented by Chang, Terlaky and Wang independently. Unlike the simplex method that follows a monotonic edge path on the feasible region, the trace of a criss-cross method is neither monotonic (with respect to the objective function) nor feasibility preserving. The main purpose of this paper is to present mathematical ideas and proof techniques behind finite criss-cross pivot methods. A recent result on the existence of a short admissible pivot path to an optimal basis is given, indicating shortest pivot paths from any basis might be indeed short for criss-cross type algorithms. The origins and the history of criss-cross methods are also touched upon.

Key words: Linear programming, quadratic programming, linear complementarity problems, oriented matroids, pivot rules, criss-cross method, cycling, recursion.

1 Introduction

Consider the following primal and dual linear programming (LP) problem pair in standard form:

$$(P) \qquad \min\{c^T x : Ax = b,\ x \geq \mathbf{0}\},$$

$$(D) \qquad \max\{b^T y : A^T y \leq c\},$$

where A is a given $m \times n$ rational matrix, c and b are n and m dimensional given rational vectors, respectively, and A^T denotes the transpose of the matrix

A. As usual, the first linear programming problem (P) is called the *primal* problem and the second problem (D) is called the *dual* problem. This pair of linear programming problems will be frequently referred to as $\mathrm{LP}(A, b, c)$.

Linear programming has been one of the most turbulent areas of applied mathematics in the last fifty years. Despite the flux of many recent approaches, Dantzig's simplex method [22] still remains to be one of the most efficient algorithms for a great majority of practical problems [5]. It is generally observed that for most cases the required number of pivot steps is a linear (and at most quadratic) function of the number of variables. This observation about the practical efficiency of the simplex method is theoretically justified by proving its polynomial behavior in the expected number of pivot steps [51,1,10,70] for various probabilistic models. On the other hand, for most variants (specified by additional pivot rules) of the simplex method there are examples for which the number of pivot steps is exponential. For such exponential examples see e.g. [2,37,46,56,57,59,60]. To find a *polynomial pivot algorithm* [1] for linear programming, i.e. one for which the number of pivot operations is bounded by a polynomial function of the input size, or to prove that such pivot algorithms do not exist [2] seems to be very hard. In fact, to clarify if there exists such a polynomial time simplex algorithm and the closely related "Hirsch conjecture" [47] are among the most challenging open problems in linear programming and in polyhedral theory.

The simplex method successfully combines greedy concepts such as preserving feasibility and forcing monotonicity of the objective value. It utilizes fundamental geometrical and combinatorial properties of linear programming problems represented by basic solutions. A further reason for the practical efficiency of the simplex method is in the fact that it is very flexible, i.e. there are plenty of possibilities to select the pivot element during the procedure, even in case of nondegeneracy. Therefore many simplex variants [3] have been developed during the last fifty years.

In the recent years, however, the major research efforts in linear programming have been shifted first to the ellipsoid method by Khachian [43] and later, even more intensively, to interior point methods [4] initiated by the seminal paper of Karmarkar [42]. New studies on pivot-based algorithms are not receiving much attention even among researchers in the field of linear programming. Moreover,

[1] A pivot algorithm might be a polynomial simplex algorithm or more generally any finite pivot method that traverses through possibly both primal and dual infeasible basic solutions.

[2] To date no exponential example is known for Zadeh's [79] pivot selection rule.

[3] A recent survey on pivot rules is given by Terlaky and Zhang [67].

[4] For recent results on interior point methods the reader can consult the books of Roos, Terlaky and Vial [61], Wright [78], Kojima et.al. [48] and the collection of survey papers in [4,66].

a large part of the work was presented in terms of oriented matroid programming and frequently without explicitly specializing the new pivot methods for the linear programming case, thus remained unknown for researchers who are not familiar with the oriented matroid terminology.

1.1 Fundamentals

Before discussing criss-cross methods, we consider problems (P) and (D) again. A vector $x \in \mathbb{R}^n$ is called *primal feasible* if it is nonnegative and $Ax = b$, i.e. satisfies all the constraints of problem (P); a vector $y \in \mathbb{R}^m$ is called *dual feasible* if $s = c - A^T y \geq \mathbf{0}$, i.e. it satisfies all the constraints in (D). The weak duality theorem provides a sufficient condition for optimality.

Proposition 1 (Weak duality) *If $x \in \mathbb{R}^n$ is primal feasible and $y \in \mathbb{R}^m$ is dual feasible then $c^T x \geq b^T y$, where the equality is satisfied iff $x^T s = 0$.*

Corollary 2 (Complementarity) *If $x \in \mathbb{R}^n$ is primal feasible, $y \in \mathbb{R}^m$ is dual feasible and $x^T s = 0$ then x and y are primal and dual optimal respectively.*

The strong duality theorem states that if $LP(A, b, c)$ is both primal and dual feasible, then there are primal and dual feasible solutions such that $c^T x = b^T y$ (or equivalently $x^T s = 0$), implying both are optimal. Consequently this sufficient condition of optimality for a dual pair of feasible solutions is necessary, too. The least-index criss-cross method described in Section 3 will yield a very simple algorithmic proof of this fundamental result.

The complementarity condition represents a strong combinatorial character of linear programming problems that is featured by pivot algorithms. In the view of the above discussions the set of optimal solutions can be characterized as the set of solutions of the system

$$
\begin{array}{ccc}
Ax = b & & A^T y + s = c \\
& x^T s = 0 & \\
x \geq \mathbf{0} & & s \geq \mathbf{0}.
\end{array}
\tag{1}
$$

Since all pivot methods [5] generate basic solutions, their intermediate solutions satisfy both equality constraints and the complementarity condition.

[5] In this paper pivot methods refer to those algorithms which maintain only one basis at a time. There are pivot algorithms of different type, e.g. the algorithm [14] maintains two different bases at a time.

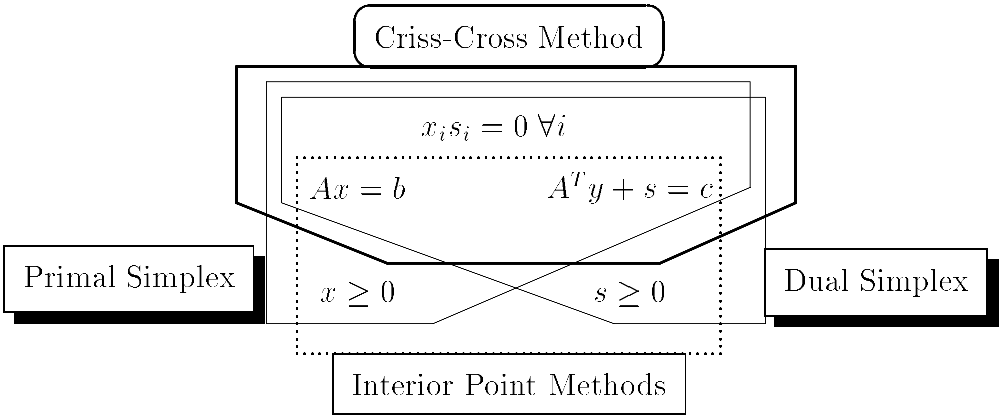

Fig. 1. Algorithms

Simplex pivot methods always require and preserve either primal or dual feasibility of the generated basic solutions. A primal (dual, respectively) simplex method is initiated by a primal (dual) feasible basic solution. If neither optimality nor unboundedness is detected then a new basis and the related basic solution is chosen in such a way that a dual (primal) infeasible variable enters (leaves) the basis and: (1) the new basis is again primal (dual) feasible; (2) the new basis differs exactly by one element from the old one, i.e. the new basis is a neighbor of the old one. Feasibility of the basic solution is preserved throughout and the primal (dual) objective function value is monotonically decreasing (increasing) at each basis exchange. To produce a primal or dual feasible basic solution is a nontrivial task. It requires the solution of another LP problem, the so-called *first phase* problem.

If no feasibility (nonnegativity) condition is forced to hold or forced to be preserved, then we can talk about *criss-cross methods*. A *finite criss-cross method* goes through some (possibly both primal and dual infeasible) basic solutions until either primal infeasibility, dual infeasibility or optimality of the LP is detected. This procedure can start with *any* basic solution and solves the linear programming problem in **one phase** in a finite number of pivot steps. The least-index criss-cross method (to be presented in Section 3) is perhaps the simplest such algorithm and gives rise to a very short algorithmic proof of the LP strong duality theorem (see Theorem 9).

Contrary to pivot methods where complementarity is maintained throughout the solution process, intermediate solutions generated by *interior point methods* typically satisfy all equality and nonnegativity conditions but complementarity. Complementarity is attained just at termination. Thus interior point methods do not make explicit use of the combinatorial features of linear programming. Figure 1 illustrates the requirements of the different methods.

Another fundamental property in linear programming is that the primal and dual feasible directions are in orthogonal subspaces i.e., the subspaces $\{x : Ax = \mathbf{0}\}$ and $\{s : s = A^T y\}$ are orthogonal. This *orthogonality* in a slightly extended form is fully utilized not only in pivot methods but in interior point methods as well. Certain "sign" properties of orthogonal subspaces furnish the base for a combinatorial abstraction [8,29] and for deeper understanding of origins and perspectives of combinatorial pivot algorithms.

The main purpose of this paper is to present mathematical ideas and proof techniques behind finite criss-cross pivot methods, and to give a fresh view on pivot algorithms.

The paper is organized as follows. At the end of this section, we present a (non-standard but useful) matrix notation which we use throughout the paper. In Section 2 termination criteria are discussed and the concept of admissible pivots is introduced. In Section 3 we present the least-index criss-cross method for linear programming problems. Based on the finiteness result a simple constructive proof of the LP strong duality theorem is given. Some discussions on the flexibility of the least-index criss-cross method and notes on worst case behavior are included as well. In Section 4 recent results on the shortest pivot sequence to an optimal basis is presented. In Section 5 some conclusions follow and further research problems are given. Finally in Section 6 the origins and the history of criss-cross methods are given.

1.2 Matrix Notation

Here we present some basic notations and definitions for linear systems.

For finite sets L and E, an $L \times E$ *matrix* is an array of doubly indexed numbers or variables

$$A = (a_{ij} : i \in L, j \in E)$$

where each member of L is called a *row index,* each member of E is called a *column index* and each a_{ij} is called a *component*. For $R \subseteq L$ and $S \subseteq E$, the $R \times S$ matrix $(a_{rs} : r \in R, s \in S)$ is called a submatrix of A, and will be denoted by A_{RS}. We use simplified notations like, A_R for A_{RE}, $A_{.S}$ for A_{LS}, A_i for $A_{\{i\}E}$, and $A_{.j}$ for $A_{L\{j\}}$. Also, for a positive integer m, we use expressions such as $m \times E$ *matrix* to mean an $L \times E$ matrix with the usual index set $L = \{1, 2, \ldots, m\}$. Thus, our matrix notation is simply an extension of the usual matrix notation, and this extension enables us to describe pivot algorithms in a simple and rigorous way.

Let $m = |L|$ and assume $\mathrm{rank}(A) = m$. Consider the homogeneous system of linear equalities:[6]

$$A\,x \;=\; \mathbf{0}, \tag{2}$$

where x is an unknown vector in $\mathbb{R}^E$. The system can be written alternately as

$$\sum_{j\in E} a_{ij}x_j \;=\; \mathbf{0} \quad \forall i \in L. \tag{3}$$

A subset B of E is said to be a *basis* if $|B| = m$ and the square submatrix $A_{\cdot B}$ is nonsingular. The complement $E \backslash B$ of a basis B is called a *nonbasis* and denoted by N. When B is a basis, we denote by $(A_{\cdot B})^{-1}$ the $B \times L$ matrix U such that the product $UA_{\cdot B}$ is the $B \times B$ identity matrix $I^{(B)}$, i.e., U is the left inverse. Since we consider matrix rows and columns to be indexed by sets, for two matrices to be multipliable, the column index set of the first matrix must be equal to the row index set of the second matrix.

Let B be any basis. The *dictionary* $D = D(B)$ of the basis B is the $B \times N$ matrix $-UA_{\cdot N} = -(A_{\cdot B})^{-1}A_{\cdot N}$. It is important to note that the dictionary represents the system of linear equations, equivalent to (2), given by

$$x_B = Dx_N, \tag{4}$$

which can also be written as

$$x_i = \sum_{j\in N} d_{ij}x_j \quad \forall i \in B. \tag{5}$$

Here the d_{ij}'s denote the components of D. For any specific nonbasic element $q \in N$, the unique solution $\overline{x}$ to the system (2) with $\overline{x}_q = 1$ and $\overline{x}_j = 0$ for all $j \in N \setminus \{q\}$ is called the *basic solution* (associated with B and q), and denoted by $x(B, q)$. Once the dictionary is given, the basic solution can easily be read: $x(B, q)_B = D_{\cdot q}$ and $x(B, q)_{N\setminus\{q\}} = \mathbf{0}$.

For those who are not familiar with the notion of dictionary (first introduced in [62] and elaborated in [15]), we note that the LP tableau matrix (see, e.g. [22]) $T = T(B)$ is related to the dictionary $D = D(B)$ by $T = [I^{(B)}\ \ -D]$.

For any $p \in B$ and $q \in N$ with $d_{pq} \neq 0$, the set $B - p + q$ is again a basis, and the replacement of B by $B - p + q$ is called the *pivot operation on* (p, q). Here, $-p$ and $+q$ are a short form of single-element deletion $\backslash\{p\}$ and addition $\cup\{q\}$, respectively.

[6] Such linear systems arise in linear programming as one introduces a new variable to homogenize the right-hand-side. The formal definition will be given in Section 3.

We define two vector subspaces of $\mathbb{R}^E$ associated with the matrix A:

$$\mathcal{N}(A) = \{x \in \mathbb{R}^E \ : \quad Ax = \mathbf{0}\} \tag{6}$$

$$\mathcal{R}(A) = \{s \in \mathbb{R}^E \ : s = A^T y, \quad y \in \mathbb{R}^L\}, \tag{7}$$

where $\mathcal{N}(A)$ is the *null space* of A and $\mathcal{R}(A)$ is the *row space* of A. When a basis B for $A\,x = \mathbf{0}$ is given, these two spaces are given symmetrically as

$$x \in \mathcal{N}(A) \iff x_B = Dx_N \tag{8}$$

$$s \in \mathcal{R}(A) \iff s_N = -D^T s_B. \tag{9}$$

This orthogonality is illustrated[7] in Figure 2. Consequently any dictionary

x	Dx_N	x_N
s	s_B	$-D^T s_B$

Fig. 2. Orthogonality of $x \in \mathcal{N}(A)$ and $s \in \mathcal{R}(A)$.

represents a dual pair of orthogonal subspaces in a very compact form. Often we have to prove certain properties of subspaces and these duality relations (8) and (9) give us a way to avoid proving two statements separately which are dual to each other.

Remark 3 (Dualization) *We use the convenient term* **dualization** *to mean exchanging the dictionary D with $-D^T$. Via dualization, one can induce a correct statement on the dual space $\mathcal{R}(A)$ by dualyzing and interpreting a proven statement on the primal space $\mathcal{N}(A)$. Although there is some asymmetry in the representation of the two spaces $\mathcal{N}(A)$ and $\mathcal{R}(A)$, we are concerned only with properties of these spaces as vector subspaces and thus it is of no importance.*

Clearly the vector $x(B,q)$, defined as $x(B,q)_B = D_{\cdot q}$, $x(B,q)_q = 1$ and $x(B,q)_{N-q} = \mathbf{0}$, is in $\mathcal{N}(A)$. Given a basis B and a vector s_B there is a unique vector $s \in \mathcal{R}(A)$ of the dual system satisfying $s_N = A^T_{\cdot N}(A^{-1}_{\cdot B})^T s_B = -D^T s_B$ and it is associated with the dual basis N. We define the dual basic solution for $p \in B$ and denote it by $s(N,p)$ if $s(N,p)_p = 1$, $s(N,p)_{B-p} = \mathbf{0}$ and $s(N,p)_N = (-D^T)_{\cdot p} = (-D_{\cdot p})^T$, i.e., the dual basic vector is represented in the pth row of D. Due to the orthogonality of the row and null spaces the above defined vectors $x(B,q)$ and $s(N,p)$ are orthogonal. One also realizes that $x(B,q)_p = -s(N,p)_q = d_{pq}$.

[7] For space efficiency here and in the subsequent figures we will draw all vectors as if they were row vectors.

2 Termination and admissible pivots

2.1 Terminal dictionaries

Considering the notation introduced in Section 1.2 for the primal-dual LP problems (P) and (D) we define the matrix

$$\overline{A} = \left(\begin{array}{c|c|c} c^T & 1 & 0 \\ \hline A & \mathbf{0} & -b \end{array} \right) \tag{10}$$

representing the orthogonal linear subspaces related to (P) and (D) respectively. Clearly the assumption $\operatorname{rank}(A) = m$ implies $\operatorname{rank}(\overline{A}) = m + 1$. We define two index sets E and $\overline{E}$ as $\{1, \ldots, n\}$ and $\{1, \ldots, n, f, g\}$,respectively, where f is associated with the unit vector (i.e. closely related to the objective vector c of the LP problem) and g is associated with the last column vector i.e. to the right-hand-side vector b. Both f and g are considered special symbols not in E and the matrix $\overline{A}$ is an $(m + 1) \times \overline{E}$ matrix.

Now it is easy to see that the primal problem (P) and the dual problem (D) of $\mathrm{LP}(A, b, c)$ are equivalent to the following formulation which we use in the sequel:

$$(\overline{P}) \qquad \max\{x_f : x \in \mathcal{N}(\overline{A}),\ x_g = 1 \text{ and } x_j \geq 0 \ \ \forall j \in E\}, \tag{11}$$

$$(\overline{D}) \qquad \max\{\, s_g : s \in \mathcal{R}(\overline{A}),\ s_f = 1 \text{ and } s_j \geq 0 \ \ \forall j \in E\}. \tag{12}$$

In this formulation, the primal problem and the dual problem are completely symmetrical. In particular, this suggests that any LP problem can be cast into the form (V, f, g) where V is a given vector subspace of $R^{\overline{E}}$. Its dual is $(V^{\perp}, g, f)$ which is obtained by taking the orthogonal subspace and interchanging the objective f and the right-hand-side g. This form represents a very strong relation between the duality of linear programming and the orthogonal duality of vector subspaces.

Comparing with the original formulation, the variable x_f is set to the negative of the original objective function $c^T x$, while the dual objective variable s_g represents the original dual objective function $b^T y$. Thus the weak duality condition Corollary 2 reads:

Proposition 4 (Symmetric form of weak duality) *For any pair (x, s) of primal-dual feasible solutions*

$$x_f + s_g = -\sum_{j=1}^{n} x_j s_j \leq 0, \tag{13}$$

and they are both optimal if the equality holds above:

$$\sum_{j=1}^{n} x_j s_j = 0. \tag{14}$$

A basis $\overline{B}$ with $f \in \overline{B} \not\ni g$ is said to be a *basis of the linear programming problem* or an *LP basis* for short. The notation $B = \overline{B} - f$ will also be used. One says that a basis is *feasible* if $x(\overline{B}, g)_B = D(\overline{B})_{B\{g\}} \geq \mathbf{0}$ and *dual feasible* if for $\overline{N} = \overline{E} - \overline{B}$, $N = \overline{N} - g$ one has $s(\overline{N}, f)_N = -D_{\{f\}N} \geq \mathbf{0}$. One sees that by definition the vectors $x(\overline{B}, g)$ and $s(\overline{N}, f)$ are feasible solutions for the problems (P) and (D) respectively. A basis is said to be *optimal* if it is both primal and dual feasible. A basis $\overline{B}$ is said to be *primal inconsistent* if there exists $p \in B$ such that $d_{pg} < 0$ and $D_{pN} \leq \mathbf{0}$, and *dual inconsistent* if there exists $q \in N$ such that $d_{fq} > 0$ and $D_{Bq} \geq \mathbf{0}$. We will call a basis (a dictionary) *terminal* if it is either optimal, primal or dual inconsistent. The sign structure of optimal, primal and dual inconsistent dictionaries are illustrated in Figure 3. Here and for the subsequent figures, we indicate the positive, nonnegative, negative, nonpositive and zero components by $+, \oplus, -, \ominus, 0$, respectively. One can easily verify the following:

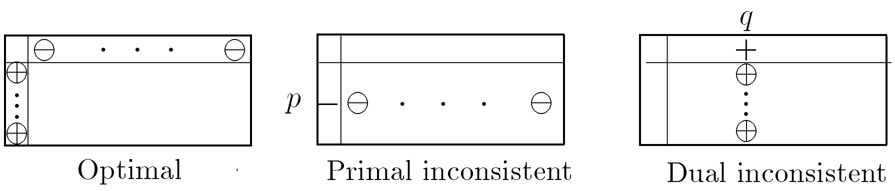

Fig. 3. Terminal tableau structures.

Proposition 5 *Let an LP basis $\overline{B}$ be given.*

(i) If $\overline{B}$ is optimal then the associated basic solutions are primal and dual optimal solutions.
(ii) If $\overline{B}$ is primal inconsistent then the primal problem is infeasible.
(iii) If $\overline{B}$ is dual inconsistent then the dual problem is infeasible.

2.2 Admissible pivots

A variable index $i \in E$ is said to be *primal* (*dual*) *feasible* at an LP basis if the associated value in the current primal (dual) basic solution is nonnegative, i.e. either $i \in N$ or $i \in B$ and $d_{ig} < 0$ (either $i \in B$ or $i \in N$ and $d_{fi} > 0$).

A natural and simple requirement that one can ask from a pivot selection is the following: when a primal (dual) infeasible variable is selected for a pivot then, after the pivot, both pivot variables should become primal (dual) feasible. Such pivots locally aim to improve the infeasibility status of the solution and are described here.

For $p \in B$ and $q \in N$ with $d_{pq} \neq 0$, a pivot on (p, q) is said to be *admissible* if either (I) $d_{pg} < 0$ and $d_{pq} > 0$ or (II) $d_{fq} > 0$ and $d_{pq} < 0$. See Figure 4. The

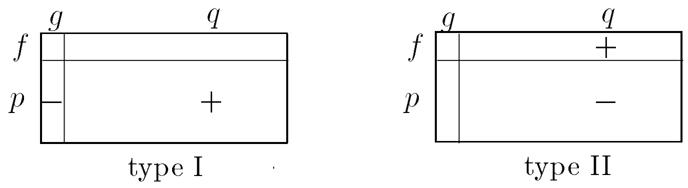

Fig. 4. Two types of admissible pivots for LP

reader can easily verify the following.

Proposition 6 *After a pivot of type I, both of p and q become primal feasible. After a pivot of type II, both of p and q become dual feasible.*

3 Finite Criss-Cross Methods

3.1 Algorithm

Now we are ready to describe the least-index criss-cross method. This criss-cross method is a purely combinatorial pivoting method for solving LP problems, it uses admissible pivots and traverses through different (possibly both primal and dual infeasible) bases until the associated basic solution is optimal, or an evidence of primal or dual infeasibility is found. It assumes that the variables are linearly ordered.

Least-Index Criss-Cross Method

Initialization
    let $\overline{B}$ be an initial basis with $f \in \overline{B} \not\ni g$;
while true **do**
    if $d_{Bg} \geq 0$ and $d_{fN} \leq 0$ **then**
(I)   **stop**: the current solution solves LP(A, b, c);
    else
        $p := \min\{i \in B : d_{ig} < 0\}$;
        $q := \min\{j \in N : d_{fj} > 0\}$;
        $r := \min\{p, q\}$;
        if $r = p$ **then**

if $d_{pN} \leq \mathbf{0}$ **then**

(II) **stop**: LP(A, b, c) has no primal feasible solution;

else

let $q := \min\{j \in N : d_{pj} > 0\}$;

endif

else (i.e. $r = q$)

if $d_{Bq} \geq \mathbf{0}$ **then**

(III) **stop**: LP(A, b, c) has no dual feasible solution;

else

let $p := \min\{i \in N : d_{iq} < 0\}$;

endif

endif

perform a pivot: $\overline{B} := \overline{B} - p + q$;

endwhile

end.

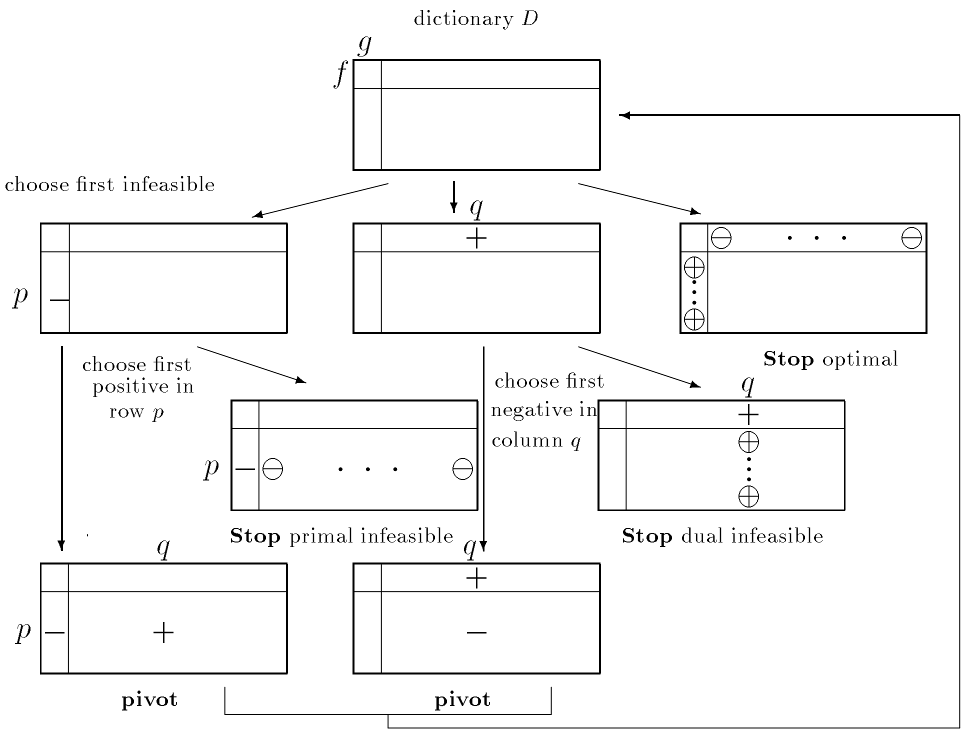

Fig. 5. Scheme of the least-index criss-cross method

See Figure 5 for a visual description of the least-index criss-cross method. It is worthwhile to mention some nice characteristics:

- It can start with any basis and thus is of only one phase.
- No ratio test is used to preserve feasibility, only the signs of components in a dictionary and a prefixed ordering of variables determine the pivot selection.

Proposition 5 shows that at termination (I) the vectors $x(\overline{B}, g)$ and $s(\overline{N}, f)$ are primal and dual optimal for LP(A, b, c), respectively. Further, primal and dual infeasibility at the cases (II) and (III) follows from the same proposition. The only thing which remains to be shown is the finiteness.

3.2 Finiteness

Directly or indirectly most (if not all) finiteness proofs of the least-index criss-cross method rely on some fundamental proposition on "almost terminal" dictionaries. First we present this proposition and a simple proof.

A dictionary is called *almost terminal* (*with respect to a constraint index* $k \in E$) if it is not terminal but by discarding a single row or column (indexed by k) it becomes terminal. Observe that we have four structurally different almost terminal dictionaries, see Figure 6.

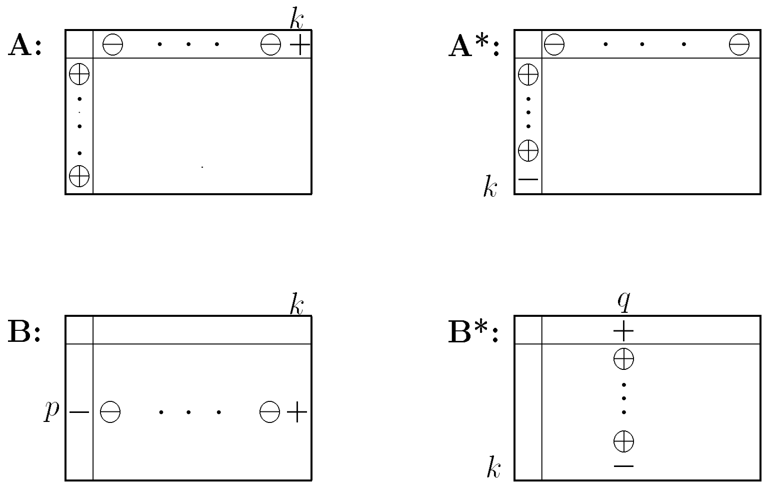

Fig. 6. Almost terminal dictionaries.

Proposition 7 *Consider a linear programming problem LP*(A, b, c) *and fix an element* $k \in E$*. Then at most one type of the four almost terminal dictionaries* **A**, **B**, **A*** *and* **B*** *with respect to* k *may exist.*

One can prove Proposition 7 in many different ways. Here we shall give two proofs. The first one makes use of the notion of redundancy of constraints. This proof, which appeared first in [27], has never been published before.

For the first proof we need one definition. An inequality or equality constraint in an LP problem is said to be *redundant* if the restriction is implied by other constraints. Thus any redundant constraint can be removed from the problem without changing the set of feasible solutions.

PROOF. (of Proposition 7.) We will show that no two different types can coexist. There are six pairs of cases to consider. We denote by **(A A*)** to mean the case that both types **A** and **A*** exist simultaneously, etc.

Let us first observe the following basic facts:

(a) If there exists a basis of type **A** then
- there is a primal feasible solution x with $x_k = 0$, and
- the constraint $s_k \geq 0$ is nonredundant for (D).

(a*) If there exists a basis of type **A*** then
- there is a dual feasible solution s with $s_k = 0$, and
- the constraint $x_k \geq 0$ is nonredundant for (P).

(b) If there exists a basis of type **B** then
- the constraint $x_k \geq 0$ is redundant for (P), and
- no primal feasible solution x with $x_k = 0$ exists.

(b*) If there exists a basis of type **B*** then
- the constraint $s_k \geq 0$ is redundant for (D), and
- no dual feasible solution s with $s_k = 0$ exists.

All claims in **(a)** and **(a*)** are straightforward from the associated basic solutions. To see **(b)**, we look at the pth equation in a type **B** dictionary: $x_p = d_{pg} + \sum_{j \in N-k} d_{pj} x_j + d_{pk} x_k$. Since $d_{pg} < 0$, $d_{pk} > 0$ and $d_{pj} \leq 0$ for all $j \in N - k$, the nonnegativity of the variables in this equation excluding x_k forces x_k to be strictly positive and thus **(b)**. By dualization (or direct interpretation), **(b*)** follows too.

The claims above immediately eliminate the possibilities of simultaneous existence of all pairs of almost terminal dictionaries, except for the case **(A A*)**, as easily verified with the following table:

Type A:	**Type A*:**
$\exists$ primal feasible x with $x_k = 0$.	$\exists$ dual feasible s with $s_k = 0$.
$s_k \geq 0$ is nonredundant for (D).	$x_k \geq 0$ is nonredundant for (P).
Type B:	**Type B*:**
$x_k \geq 0$ is redundant for (P).	$s_k \geq 0$ is redundant for (D).
$\nexists$ primal feasible x with $x_k = 0$.	$\nexists$ dual feasible s with $s_k = 0$.

To prove that the remaining case **(A A*)** is impossible, assume we have a dictionary D of type **A** and a dictionary D' of type **A***. Let $x = x(\overline{B}, g)$ and $x' = x(\overline{B}', g)$. By the sign patterns of these dictionaries, it is easy to see

(*) both x and x' are optimal for the LP problem $\overline{\text{LP}}$ obtained from the primal LP by reversing the inequality in $x_k \geq 0$ (i.e. $x_k \leq 0$).

Moreover, observing the equation represented by the fth row of D, since $d_{fk} > 0$ and $d_{fj} \leq 0$ for all $j \in B - k$, every feasible solution to $\overline{\text{LP}}$ with a negative kth component has an objective value strictly less than the optimal value d_{fg}. Thus no optimal solution to $\overline{\text{LP}}$ can have a negative kth component. Since $x'_k < 0$, we have a contradiction. $\square$

Now we give the second proof which makes use of orthogonality of two vector subspaces $\mathcal{N}(\overline{A})$ and $\mathcal{R}(\overline{A})$.

PROOF. (Second proof of Proposition 7.) We have six pairs of cases to consider. We use the same notation such as **(A A*)** to mean the case that both types **A** and **A*** exist simultaneously. In this proof a dictionary and the associated basis for case **A** or **B** will be denoted by $D, \overline{B}, \overline{N}$, and for case **A*** or **B*** by $D', \overline{B}', \overline{N}'$. The case **(A A*)** is the most involved, thus we begin first with the easier cases.

(B B*): The vectors $s(\overline{N}, p)$ and $x(\overline{B}', q)$ are in the orthogonal subspaces $\mathcal{R}(\overline{A})$ and $\mathcal{N}(\overline{A})$ respectively. It is easy to see from the sign patterns of dictionaries that their inner product is positive (see the following figure), contradicting to the orthogonality. Thus this case cannot occur.

$$\begin{array}{r|c|c|c|c|c|c|c|c|}
 & f & g & & & & & & k \\
s(\overline{N}, p) & 0 & + & \oplus & \cdot & \cdot & \cdot & \oplus & - \\
x(\overline{B}', q) & + & 0 & \oplus & \cdot & \cdot & \cdot & \oplus & -
\end{array}$$

Fig. 7. Case **(B B*)**

(A B*): The vectors $s(\overline{N}, f)$ and $x(\overline{B}', q)$ are in the orthogonal subspaces $\mathcal{R}(\overline{A})$ and $\mathcal{N}(\overline{A})$ respectively. Clearly their inner product is positive as shown in the figure below, a contradiction. Thus this case cannot occur.

$$\begin{array}{r|c|c|c|c|c|c|c|c|}
 & f & g & & & & & & k \\
s(\overline{N}, f) & 1 & & \oplus & \cdot & \cdot & \cdot & \oplus & - \\
x(\overline{B}', q) & + & 0 & \oplus & \cdot & \cdot & \cdot & \oplus & -
\end{array}$$

Fig. 8. Case **(A B*)**

(B A*): By dualization, this case reduces to the case **(A B*)**, and thus this case cannot occur.

(A A*): This is a slightly more involved case. The vectors $s(\overline{N}, f)$ and $s(\overline{N}', f)$ are in the row space $\mathcal{R}(\overline{A})$ and the vectors $x(\overline{B}, g)$ and $x(\overline{B}', g)$ are in the null space $\mathcal{N}(\overline{A})$. See the following figure for their sign structures.

	f	g				k
$s(\overline{N}, f)$	1		$\oplus$	$\cdots$	$\oplus$	$-$
$x(\overline{B}, g)$		1	$\oplus$	$\cdots$	$\oplus$	0
$s(\overline{N}', f)$	1		$\oplus$	$\cdots$	$\oplus$	0
$x(\overline{B}', g)$		1	$\oplus$	$\cdots$	$\oplus$	$-$

Fig. 9. Case **(A A*)**

The vectors $[s(\overline{N}, f) - s(\overline{N}', f)]$ and $[x(\overline{B}', g) - x(\overline{B}, g)]$ are orthogonal as well. Observe their product is positive because (1) the f component is zero in the first vector; (2) the g component is zero in the second vector; (3) the k component is negative in both vectors; (4) for each of the remaining components the product is nonnegative. This contradicts to the fact that they are orthogonal. Thus the types **A** and **A*** cannot exist simultaneously.

(A* B*): This case is similar to the case **(A B*)**.

(A B): This case is dual to **(A* B*)** and also similar to the case **(B A*)**. □

Now we prove the finiteness of the least-index criss-cross method.

Theorem 8 *The least-index criss-cross method is finite.*

PROOF. Since the number of bases is finite, it is enough to show that each basis might be visited at most once by the algorithm. The proof is by contradiction. Assume that a basis is visited twice. Then the algorithm generates a cycle, i.e. starting from this basis, after a certain number of steps the same basis is reached again. Without loss of generality we may also assume that this cycling example is minimal and $E = \{1, \ldots, n\}$. This implies each variable in E, in particular n, must enter and leave a basis during the cycle.

There are two situations when n as a nonbasic element might enter a basis. The first case **A** is when the current dictionary is almost optimal with respect to n, and the second case **B** is when some basic variable, say p, is is primal infeasible and d_{pn} determines the only admissible pivot in row p. Similarly, there are only two cases in which n as a basic element might leave a basis. The first case **A*** is when the current dictionary is almost optimal with respect to n, and the second case **B** is when some nonbasic variable, say q, is is dual infeasible and d_{nq} determines the only admissible pivot in column q. Thus at least one case of the four possible combinations **(A A*)**, **(A B*)**, **(B A*)** and **(B B*)** must occur. On the other hand, the described cases **A**, **B**, **A*** and **B*** coincide with the four almost terminal dictionaries of Figure 6 for $k = n$. By Proposition 7, none of **(A A*)**, **(A B*)**, **(B A*)** and **(B B*)** can occur, a contradiction. □

One of the most important consequences of the finiteness of the least-index criss-cross method is the *strong duality theorem* of linear programming. We believe this is one of the simplest algorithmic proofs of this fundamental result.

Theorem 9 [Strong duality] *Exactly one of the following two cases occurs for any linear programming problem $LP(A, b, c)$.*

- *At least one of the primal problem (P) and the dual problem (D) is infeasible.*
- *Both problems have an optimal solution and the optimal objective values are equal.*

PROOF. By using Gaussian elimination, one finds either the inconsistency of $Ax = b$ (thus identifies primal infeasibility), or an LP basis in finite steps, possibly discarding some redundant equations. For the latter case, apply the least-index criss-cross method with this initial LP basis. By Theorem 8, it terminates with one of the three terminal dictionaries of Proposition 5 (see also Figure 3), hence either an optimal solution pair is obtained or infeasibility is detected either from the primal or from the dual side. $\square$

It is worthwhile to mention another simple proof of the strong duality theorem due to Edmonds [26]. It involves a new way of solving an LP problem by considering a sequence of feasibility problems with an extra inequality expressing the objective function bounded by certain constant. By a clever update of the constant the algorithm finds the optimal value by solving finitely many feasibility problems. He also proposes a combinatorial pivot algorithm to solve feasibility problems whose correctness can be proved by a simpler version of Proposition 7.

The following corollaries are based on different interpretations of the least-index criss-cross method and this way alternative proofs can be derived. The first corollary is based on a recursive interpretation of the algorithm. This recursive interpretation and the finiteness proof based on it can be derived from the results due to Bland and Jensen [7,8,40] and can be found in Väliaho's paper [72].

Corollary 10 (Recursive interpretation.) *As performing the least-index criss-cross method at each pivot one can make a note of the larger of the two indices $r = \max\{p, q\}$ that entered or left the basis. In this list, an index must be followed by another larger one before the same index occurs anew.*

PROOF. *At a given basis, by using the predefined ordering of the indices in the dictionary, we identify first the largest index $r - 1$ for which all the variables $i \leq r - 1$ are both primal and dual feasible. If $r - 1 = n$, the problem* *is optimally solved. Else, r is primal or dual infeasible. Let $t-1$ be the largest index for which $d_{rj} \leq 0$ for all $j \leq t-1$ or $d_{ir} \geq 0$ for all $i \leq t-1$, respectively. If $t-1 = n$ then primal or dual infeasibility is detected. Else, let $k = \max\{r, t\}$. Then we have a terminal dictionary for the subproblem involving the variables $i \leq k-1$ and an almost terminal dictionary for the subproblem involving the variables $i \leq k$. The new pivot involves the index k as the larger one of the two pivot indices. This happens at one of the four situations illustrated at Figure 6 if we look at the subproblem involving only the indices $i \leq k$. When the same index k occurs anew in the list, then an index higher than k must have been involved in a pivot want else at the new occurrence of the index k another dictionary from Figure 6 for the subproblem $i \leq k$ would occur that is not possible by Proposition 7.* □

The recursive interpretation is becoming apparent when one notes that the size of the solved subproblem (the subproblem for which a terminal dictionary is obtained) is monotonically increasing. The third finiteness proof is based on the technique developed by Edmonds and Fukuda [29] and adapted by Fukuda and Matsui [32] to the case of the least-index criss-cross method.

Corollary 11 (Lexicographically increasing list.) *Let $u \in \mathbb{R}^n$ be a binary vector set initially to be the zero vector. In applying the algorithm let $r = \max\{p, q\}$ be the larger of the two indices p that entered or q that left the basis. At each pivot update u as follows: let $u_r = 1$ and $u_i = 0$ $\forall i < r$. The remaining components of u stay unchanged. Then at each step of the least-index criss-cross method the vector u strictly increases lexicographically, thus the method terminates in a finite number of steps.*

PROOF. This follows immediately from Corollary 10. □

In all three finiteness proofs above, the main tool is Proposition 7. Note that the finiteness of Bland's least-index rule [8] is an easy consequence of this proposition as well.

3.3 Flexibility

Once an ordering of the index set is chosen, the pivot sequence of the least-index criss-cross method is uniquely determined. This way the least-index criss-cross method seems to be completely rigid. Having a closer look at the finiteness proofs one can see the essential mechanism to guarantee the finiteness and there is a great deal of hidden flexibility. This can be easily seen from the last convergence theorem, Corollary 11. There one can easily analyze what kind of reordering of the index list is allowed while the strict lexicographic increase of the indicator vector u is preserved.

A *zero interval* of an indicator vector u is a set of consecutive indices i in E such that $u_i = 0$. At beginning the set E is itself a zero interval. It is very easy to see that the variable ordering within any zero interval is not important for the finiteness. This implies that one can reorder the variables within any zero intervals as one likes. These reorderings do not change the vector u, and do not change the position of the components equal to one in u.

It should be noted that one can exploit the variable reordering even further. The idea is to "forget" some pivot history coded in the u vector if one can gain more. An extreme case is when it is better to reorder all the variables by forgetting all the history and starting all over from scratch. Such an idea was presented in [26,31].

Both the recursive interpretation and the flexibility of pivot selection in the least-index criss-cross method make it possible to develop other finite variants. Such finite criss-cross methods, which do not rely on a fixed index ordering, can be developed based on the finite simplex rules presented by Zhang [80]. Using either of the above presented two Corollaries 10 or 11 the reader can easily verify the finiteness of the following two criss-cross methods [81].

– *First in last out (FILO).* – First, choose a primal or dual infeasible variable that has changed its basis/nonbasis status most recently. Then choose a variable in the selected row or column so that the pivot is admissible and that has changed its basis/nonbasis status most recently. When more than one candidates exist with the same pivot age, break tie as you like (e.g. randomly).
– *Most often selected variable.*– First, choose a primal or dual infeasible variable that has changed its basis/nonbasis status most frequently. Then choose a variable in the selected row or column so that the pivot is admissible and that has changed its basis/nonbasis status most frequently. When more than one candidates exist with the same pivot frequency, break tie as you like (e.g. randomly).

3.4 Exponential and average behavior

The worst case exponential behavior of the least-index criss-cross method was studied by Roos [60]. Roos' exponential example is a variant of the Klee-Minty [46] cube. In this example the starting solution is the origin defined by a feasible basis, the variables are ordered so that the finite criss-cross method follows a simplex path, i.e. without making any ratio test feasibility of the starting basis is preserved. Another exponential example was presented by Fukuda and Namiki [33] for linear complementarity problems.

Contrary to the worst case behavior, to date not much is known about the expected number of pivot steps required by criss-cross methods to solve LP problems. The only notable result was obtained by Fukuda and Namiki [33]. They averaged the length of all possible pivot sequences on an exponential example for linear complementarity problems by considering all possible orderings of the variables. They proved that for some class of problems where the longest criss-cross path is exponential, the average length of all possible criss-cross paths is exactly the dimension of the problem space.

4 Best-case Analysis of Admissible Pivot Methods

As it is the case for many simplex algorithms, the least-index criss-cross method discussed in Section 3 is not a polynomial-time algorithm, see [60]. A question we would like to ask is as to whether there exists a polynomial criss-cross method? Unfortunately we have no reasonable ways to answer this question at this moment. However a recent work of Fukuda-Lüthi-Namiki [30] shows the existence of a short sequence of admissible pivots (introduced in Section 2.2) of length at most m to an optimal basis from any given basis.

In this section we shall present this result (in a slightly weaker form) which indicates a potential direction for us to look for polynomial time pivot algorithms. It is noteworthy that we do not know of any such result for feasibility preserving algorithms, and in fact, the maximum length of shortest feasibility-preserving pivot sequences between two feasible bases is not known to be bounded by a polynomial in the size of the associated LP.

4.1 Short pivot sequences

Throughout, we assume that an LP basis is given. Finding one LP basis (or proving that none exists) can easily be done by Gaussian elimination. The *goal* of a pivot algorithm is to find a terminal basis by a sequence of pivot operations starting with a given LP basis.

Let B be an LP basis, i.e. a basis with $f \in B \not\ni g$. Recall that for $p \in B - f$ and $q \in N - g$ with $d_{pq} \neq 0$, a pivot on (p, q) is said to be *admissible* if either (I) $d_{pg} < 0$ and $d_{pq} > 0$ or (II) $d_{fq} > 0$ and $d_{pq} < 0$ (see Figure 4).

Admissible pivots are very natural elementary operations to be used in any pivot algorithms by the following reasons.

Proposition 12 *Let B be any LP basis. Then the following statements hold.*

(a) *If B is not terminal, then there exists an admissible pivot at B.*
(b) *If the LP has an optimal solution, then there exists an admissible pivot at B if and only if B is not optimal.*

PROOF. The statement **(a)** is clear with the definitions. To prove **(b)**, assume that the LP has an optimal solution. This together with the LP duality theorem, Theorem 9 implies that neither a primal inconsistent nor a dual inconsistent basis exists. Thus if a basis B is not optimal, there exists an admissible pivot. The other implication is trivial. $\square$

We define an *admissible pivot method* for LP as a pivot method that only uses admissible pivots. It is not difficult to verify the following.

Remark 13 *The primal simplex method, the dual simplex method, Bland's recursive pivot method [8], Edmonds-Fukuda method [29,6] and the least-index criss-cross method [64,65,74] are all admissible pivot methods. More precisely, the primal simplex method and Edmonds-Fukuda method use admissible pivots of type II only, the dual simplex method uses admissible pivots of type I only, and both Bland's recursive method and the criss-cross methods discussed in Section 3 use admissible pivots of both type I and type II.*

Now a simple fundamental question arises.

Question: What is the length of the shortest sequence of admissible pivots from a given LP basis B to any fixed optimal basis B^*?

It turns out that this question has a rather disappointing answer. In some cases, no such sequence exists and thus the shortest length is undefined, as shown in the following example.

Example 14 *Consider the LP illustrated in Figure 10. In this example, $E = \{1, 2, 3, 4, f, g\}$, and the set of feasible solutions is the shaded triangle region. There are two optimal bases $B^* = \{1, 4, f\}$ and $B^{**} = \{1, 3, f\}$. At the dictionary for the basis $B = \{3, 4, f\}$, there is only one admissible pivot, that is at $(4, 1)$. Clearly by this pivot, one reaches the basis B^{**}. Thus, the basis B^* is not reachable from B by any admissible pivots.*

The question becomes interesting once we assume nondegeneracy. We say that an LP basis B is *nondegenerate* if $d_{ig} \neq 0$ for all basic variable $i \in B - f$. An LP basis B is said to be *dual-nondegenerate* if the associated nonbasis is

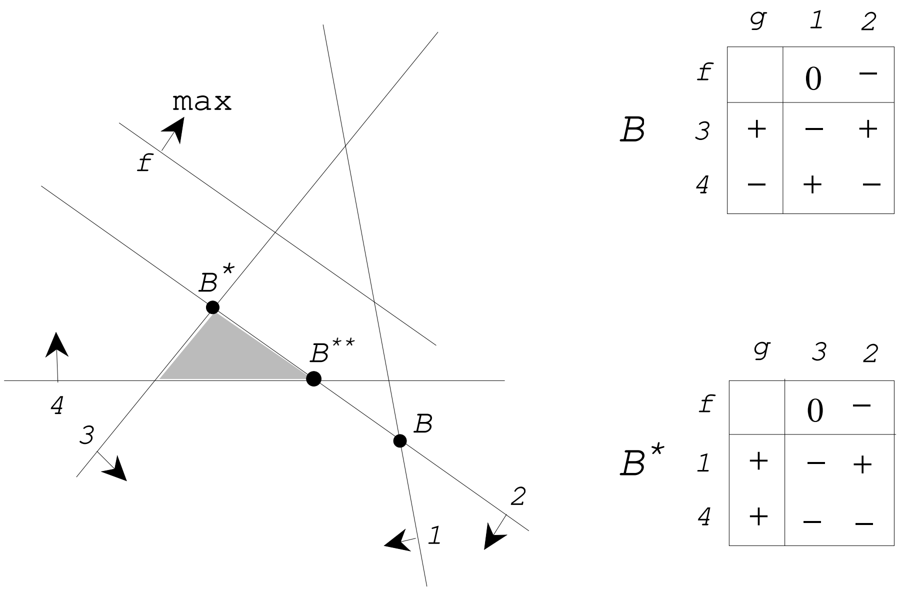

Fig. 10. B^* unreachable from B by admissible pivots

nondegenerate for the dual problem, or equivalently $d_{fj} \neq 0$ for all nonbasic variable $j \in N - g$. Finally, an LP basis is *fully nondegenerate* if it is both nondegenerate and dual-nondegenerate. One can easily prove the following.

Lemma 15 *If LP has a fully nondegenerate optimal basis, then it is the unique optimal basis.*

Our main theorem in this section is the following which is a consequence of the uniqueness lemma above.

Theorem 16 *If an LP has a fully nondegenerate optimal basis B^*, then there exists a sequence of $|B^* \backslash B|$ admissible pivots from any LP basis B.*

PROOF. Let B^* be a fully nondegenerate optimal basis for a given LP and let B be any LP basis different from B^*. Set $B_0 = B \backslash B^*$, $B_1 = B^* \cap B$, $N_0 = N \backslash N^* = B^* \backslash B$ and $N_1 = N^* \cap N$. Consider the submatrix $D' = D_{(B_0+f)(N_0+g)}$ of the dictionary $D = D(B)$. The matrix D' represents a smaller linear optimization problem LP$'$ with variables $B_0 \cup N_0$, see Figure 11.

It is easy to see that LP$'$ has a fully nondegenerate optimal basis, namely $N_0 + f$, and by Lemma 15 it is a unique optimal basis. This implies that the basis $B_0 + f$ cannot be terminal. This together with Proposition 12(a) implies that there exists an admissible pivot in D'. This transformation makes the resulting basis closer to the feasible basis B^* by one. Hence, by applying the same argument, we can arrive at B^* by a sequence of admissible transformations involving exactly $|B^* \backslash B|$ pivots. $\square$

One might suspect that this theorem is a disguised form of a simple well-

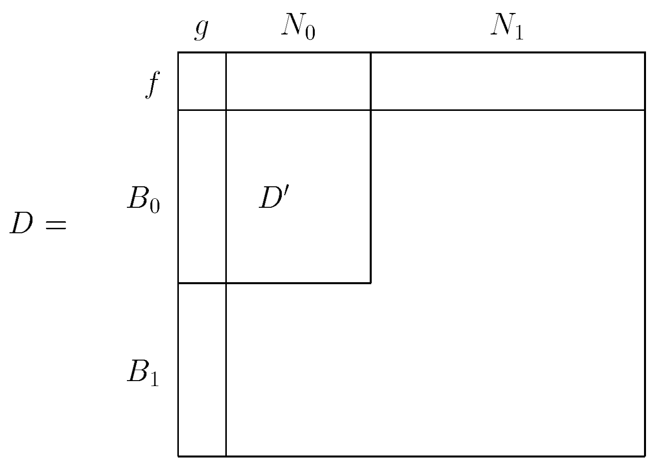

Fig. 11. A Partition of the dictionary $D = D(B)$

known lemma in linear algebra: any two bases B and B' of a matrix A can be connected by a sequence of $|B' \backslash B|$ pivot operations. As it was illustrated by Example 14, the validity of our theorem critically depends on the full nondegeneracy assumption, which has no counterpart in this lemma. Therefore we do not see any easy way to relate these two statements with similar appearances.

In order to apply the theorem to general degenerate LP's, we can use a standard technique of lexicographic (symbolic) perturbation to both the primal and the dual LP. This makes all LP bases fully nondegenerate symbolically, see [30] for details.

5 Conclusions and open problems

The goal of the present paper was to provide a good motivation to study a class of pivot algorithms which is more general than the class of simplex algorithms. This broader view on pivot methods resulted in a family of simple, finite, one-phase methods, it made possible to give simple constructive proofs for the LP strong duality theorem, and it allowed us to explore the combinatorial structures of linear optimization problems. New results on the existence of a short admissible pivot path were presented. This is a weaker analogue of the old, very difficult problem, the so-called *d-step* conjecture [47]. Some related open problems are sketched below.

The simplex method has been studied extensively by many researchers since it was invented in 1947 by Dantzig [21,23], but yet it resists the resolution of the long-standing open problem:

Open Problem 1: Is there any pivot rule for the simplex method which terminates in a number of pivots bounded by a polynomial in m and n?

Unfortunately this problem 1 seems to be too hard because the existence question:

Open Problem 2: Is there a simplex-pivot path to an optimal basis from any given feasible basis whose length is bounded by a polynomial of m and n?

appears to be already too hard. Observe that a solution of the first problem gives a solution to the second one. Furthermore, the affirmative answer to this latter simpler question would imply the solution of the famous diameter problem of convex polytopes:

Open Problem 3: Is the combinatorial diameter of a d-dimensional polytope with m-facets bounded by a polynomial in m and d?

Although some interesting results on the diameter of polytopes have been obtained recently (e.g. [41]), we must admit that we are still quite far from the solution to this "simpler" problem.

In Section 4 a positive answer to a much weaker question is presented. We have seen that if the optimal basis is fully nondegenerate, a short admissible pivot path from any basis to the optimal basis exists. This existence result raises the following more difficult question (posed by H.-J. Lüthi):

Open Problem 4: For any given LP, is there an ordering of variables for which the least-index criss-cross method finds an optimal basis from any given basis in a polynomial number of pivots?

A much simplified question, which seems to be already hard, is:

Open Problem 4': For any given LP, a given optimal basis and a given initial basis, is there an ordering of variables for which the least index criss-cross method terminates in a polynomial number of pivots?

Although the least-index criss-cross method presented in Section 3 admits an exponential example, the ultimate goal of our study is of course to answer the most basic question:

Open Problem 5: Is there a polynomial criss-cross method?

Another open problem concerns the practicality of criss-cross rules. Contrary to the many efficient implementations of simplex methods, to date no practically efficient criss-cross method is known. Thus the following question remains.

Open Problem 6: Is there a practically efficient finite criss-cross method?

Preliminary but interesting experimental results on some modifications of the least-index criss-cross method were presented in [58].

As we mentioned earlier, not much is known on the expected number of pivot steps of criss-cross methods. This is in sharp contrast to the extensive literature on the expected number of steps of simplex methods.

Open Problem 7: What is the expected number of pivots required by a finite criss-cross method to solve an LP?

Of course this is not a single question. One can pose this question with respect to many different probabilistic models and with different criss-cross algorithms.

6 Historical Notes

6.1 Origins

From the early days of LP research, people have been looking for an algorithm that avoids the two phase procedure needed in the simplex method when solving the general LP problem. Such a method was assumed to rely on the intrinsic symmetry behind the primal and dual problems (self-dual), and it should be able to start with any basic solution.

There were several attempts made to relax the feasibility requirement in the simplex method. It is important to mention Dantzig's [22] parametric self-dual simplex algorithm. This algorithm can be interpreted as Lemke's [50] algorithm for the corresponding linear complementarity problem [52]. In the sixties people realized that pivot sequences through possibly infeasible basic solutions might result in significantly shorter paths to the optimum. Moreover a self-dual one phase procedure was expected to make linear programming more easily accessible for broader public. Probably these advantages stimulated the birth of Zionts' [82,83] criss-cross method (after which the finite criss-cross methods discussed in this paper were named too).

Remark 17 (Zionts' criss-cross method) *Assuming that the reader is familiar with both the primal and dual simplex methods, Zionts' criss-cross* *method can easily be explained.*

- *It can be initialized by any, possibly both primal and dual infeasible basis. If the basis is not optimal, then there are some primal or dual infeasible variables. One might choose any of these. It is advised to choose primal and dual infeasible variables alternately if possible.*
- *If the selected variable is dual infeasible, then it enters the basis and the leaving variable is chosen among the primal feasible variables in such a way that primal feasibility of these variables is preserved.*

  If no such basis exchange is possible another infeasible variable is selected.
- *If the selected variable is primal infeasible, then it leaves the basis and the entering variable is chosen among the dual feasible variables in such a way that dual feasibility of these variables is preserved.*

  If no such basis exchange is possible another infeasible variable is selected.

If the current basis is infeasible, but none of the infeasible variables allows a pivot fulfilling the above requirements then it is proved that the problem $LP(A,b,c)$ *has no optimal solution.*

Once a primal or dual feasible solution is reached then Zionts' method reduces to the primal or dual simplex method respectively.

One attractive character of Zionts' criss-cross method is primal-dual symmetry (self-duality), and this alone differentiates itself from the simplex method. However it is not clear [8] how one can design a finite version (i.e. a finite pivot rule) of this method. Nevertheless, this is the first root of the finite criss-cross methods studied in this paper.

The other root of finite criss-cross methods was the intellectual effort to find finite variants (other than the lexicographic rule [13,24]) of the simplex method and its specialized variants. These efforts were also stimulated by studying the combinatorial structures behind linear programming. From the early seventies in several branches of the optimization theory, finitely convergent algorithms were published, and in particular *consistent labelling* [9] and pivot selection rules based on *least index ordering* [10] established a solid foundation for finite pivot algorithms.

[8] Because in some cases Zionts' method reduces to the simplex method, lexicographic perturbation or minimal index resolution is a must when one thinks about finiteness. However these do not seem to be sufficient to prove finiteness in the general case when the initial basis is both primal and dual infeasible.

[9] See Tucker's [71] consistent labelling technique in the Ford-Fulkerson flow algorithm

[10] See Murty's [55] Bard-type scheme for the P-matrix linear complementarity problem and Bland's [8] celebrated least index rule in linear programming.

The birth of oriented matroids and oriented matroid programming [8,9,28] gave an essential impulse to this research. It became clear that although the simplex method has rich combinatorial structures, some essential results like the finiteness of Bland's minimal index rule [8] cannot be proved in the oriented matroid context. In fact Edmonds and Fukuda [29] showed that it might cycle in the oriented matroid case due to the possibility of nondegenerate cycling which is impossible in linear programming.

Interesting combinatorial pivot algorithms were discovered and mainly published in the oriented matroid context. Among them are Bland's recursive algorithm [7,8], the Edmonds-Fukuda algorithm [29] (the interested reader can find variants and generalizations in e.g. [6,16,75,76,80]). All these methods in the linear case are variants of the simplex method, i.e. they preserve the feasibility of the basis. In contrast, in the case of oriented matroid programming only Todd's finite lexicographic method [68,69] preserves feasibility of the basis and therefore yields a finite simplex algorithm for oriented matroids. For a thorough survey on pivot algorithms the reader is referred to [67].

All these research results made the time ripe for the discovery of finite criss-cross methods. It is remarkable that almost at the same time, in different parts of the world (China, Hungary, USA) essentially the same result was obtained independently by approaching the problem from quite different directions.

6.2 Birth

The first finite criss-cross algorithm, which we called the *least-index criss-cross method*, was discovered independently by Chang [12], Terlaky [63,65,64] and Wang [74] A strongly related general recursion was found by Jensen [40].

In his unpublished report [12] Chang presented in 1979 this algorithm for positive semidefinite linear complementarity problems as a generalization of Murty's [55] Bard-type scheme. He was studying the role of least index resolution of degeneracy problems in Lemke's algorithm [49,50] and in variants of the principal pivoting method [18] as one solves linear complementarity problems. If one specializes the algorithm on page 59 of his long report to the linear programming case, precisely the finite, least-index criss-cross method is obtained. Unfortunately this result remained unknown until the early nineties [38,72].

At the end of 1983, Terlaky presented the least-index criss-cross algorithm for LP [63,64] and then immediately generalized in 1984 to the oriented matroid case [65]. Since then the algorithm was known by the name "finite (or convergent) criss-cross method". This name was given due to the motivation from Zionts' [82] ideas. These papers inspired a good deal of research on criss-cross methods in the last decade.

Wang's motivation was quite different. He was working on oriented matroids and tried to resolve the problems arising from possible cycling of Bland's least index simplex algorithm in oriented matroids. In 1985 Wang [74] presented his version for oriented matroid programming under the name "a finite conformal elimination free algorithm" which again turned out to be the same finite criss-cross method.

Bland and his then Ph.D. student Jensen worked on the recursive line. Their research resulted in a general recursive scheme [40]. Jensen's "generalized relaxed recursive scheme" is actually a general model for finite pivot algorithms and it is easy to see that the least-index criss-cross method can be interpreted as a concrete realization of this general scheme.

Acknowledgement

The authors are grateful to Tibor Illés, Thomas M. Liebling, Hans-Jakob Lüthi, Kees Roos and Shuzhong Zhang for reading earlier versions of this paper and making useful comments that improved the presentation of the paper considerably.

References

[1] I. Adler and N. Megiddo, A simplex algorithm whose average number of steps is bounded between two quadratic functions of the smaller dimension, *Journal of the Association of Computing Machinery* **32** (1985) 891–895.

[2] D. Avis and V. Chvátal, Notes on Bland's rule, *Mathematical Programming Study* **8** (1978) 24–34.

[3] D. Avis and K. Fukuda, A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra, *Discrete Comput. Geom.* **8** (1992) 295–313.

[4] J. Beasley (ed.), *Advances in Linear and Integer Programming*, (Oxford University Press, Oxford, England, 1996).

[5] R.E. Bixby, The simplex method: it keeps getting better, Lecture at the 14th International Symposium on Mathematical Programming, Amsterdam, The Netherlands (1991).

[6] A. Björner, M. Las Vergnas, B. Sturmfels, N. White and G. Ziegler, *Oriented Matroids*, (Cambridge University Press, 1993).

[7] R.G. Bland, New finite pivoting rules for the simplex method, *Mathematics of Operations Research* **2** (1977) 103–107.

[8] R.G. Bland, A combinatorial abstraction of linear programming, *Journal of Combinatorial Theory (Ser. B)* **23** (1977) 33–57.

[9] R.G. Bland and M. Las Vergnas, Orientability of matroids, *Journal of Combinatorial Theory, Ser. B.* **24** (1978) 94–123.

[10] K.H. Borgwardt, *The Simplex Method: A Probabilistic Analysis,* Algorithms and Combinatorics, Vol. 1. (Springer Verlag, 1987).

[11] K. Cameron and J. Edmonds, Existentially polytime theorems, *Proceedings of the DIMACS Workshop in Polyhedral Combinatorics* (1993).

[12] 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).

[13] A. Charnes, Optimality and degeneracy in linear programming, *Econometrica* **20**(1952)160–170.

[14] H.-D. Chen, P.M. Pardalos, M. A. Saunders, The simplex algorithm with a new primal and dual pivot rule, *Operations Research Letters* **16** (1994) 121–127.

[15] V. Chvátal, *Linear Programming*, (W.H. Freeman and Company, 1983).

[16] J. Clausen, A note on Edmonds-Fukuda pivoting rule for the simplex method, *European Journal of Operations Research* **29** (1987) 378–383.

[17] R.W. Cottle, Symmetric dual quadratic programs, *Quart. Appl. Math.* **21** (1963) 237-243.

[18] R.W. Cottle and G.B. Dantzig, Complementary pivot theory of mathematical programming, *Linear Algebra and Its Applications* **1** (1968) 103-125.

[19] R.W. Cottle, J.-S. Pang and V. Venkateswaran, Sufficient matrices and the linear complementarity problem, *Linear Algebra and Its Applications* **114/115** (1987) 235–249.

[20] R.W. Cottle, J.-S. Pang, and R.E. Stone, *The linear complementarity problem*, Academic Press, 1992.

[21] G.B. Dantzig, Programming in a linear structure, *Comptroller*, USAF Washington, DC, (February 1948).

[22] G.B. Dantzig, *Linear Programming and Extensions*, (Princeton University Press, Princeton N.J., 1963)

[23] G.B. Dantzig, Linear programming: The story about how it began, In: *History of Mathematical Programming* eds. A.H.G. Rinnoy Kan, L.K. Lenstra and A. Schrijver, (CWI North-Holland, Amsterdam 1991) 19–31.

[24] G.B. Dantzig, A. Orden, and P. Wolfe, The generalized simplex method for minimizing a linear form under linear inequality restraints, *Pacific Journal of Mathematics* **5**(1955) 183–195.

[25] J. Edmonds, Exact pivoting, presentation, ECCO VII, February 1994.

[26] J. Edmonds, A Helly method for linear programming, presentation, CO94, Amsterdam, April 1994.

[27] J. Edmonds and K. Fukuda, NP easy and LP theory, Cours postgrade en Recherche Opérationnelle, École Polytechnique Fédérale de Lausanne, Lausanne, May 1994.

[28] J. Folkman and J. Lawrence, Oriented matroids, *Journal of Combinatorial Theory (Ser. B)* **25** (1978) 199–236.

[29] K. Fukuda, *Oriented Matroid Programming*, Ph.D. Thesis, (Waterloo University, Waterloo, Ontario, Canada 1982).

[30] K. Fukuda, H.-J. Lüthi and M. Namiki, The existence of a short sequence of admissible pivots to an optimal basis in LP and LCP, to appear in ITOR.

[31] K. Fukuda and H.-J. Lüthi, Combinatorial maximum improvement algorithm for LP and LCP, presented at Franco-Japanese Days, Brest, France (1995).

[32] K. Fukuda and T. Matsui, On the finiteness of the criss-cross method, *European Journal of Operations Research* **52** (1991) 119–124.

[33] K. Fukuda and M. Namiki, On extremal behaviors of Murty's least index method, *Mathematical Programming* **64** (1994) 365–370.

[34] K. Fukuda and M. Namiki, Finding all common basis in two matroids, *Discrete Applied Mathematics* **56** (1995) 231–243.

[35] K. Fukuda, M. Namiki and A. Tamura, EP theorems and linear complementarity problems, *Technical Report*, Institute for Operations Research, ETH-Zentrum, CH-8092 Zürich, Switzerland (1996).

[36] K. Fukuda and T. Terlaky, Linear complementarity and oriented matroids, *Journal of the Operational Research Society of Japan* **35** (1992) 45–61.

[37] D. Goldfarb, Worst case complexity of the shadow vertex simplex algorithm, *Technical Report*, Department of Industrial Engineering and Operations Research, Columbia University, USA (1983).

[38] 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.

[39] T. Illés, Á. Szirmai and T. Terlaky, A finite Criss-Cross Method for Hyperbolic Programming, *Report No. 96–103*, Faculteit der Technische Wiskunde en Informatica, Technische Universiteit Delft, The Netherlands (1996). To appear in *European Journal of Operations Research*.

[40] D. Jensen, *Coloring and Duality: Combinatorial Augmentation Methods*, Ph.D. Thesis, (School of OR and IE, Cornell University, Ithaca, N.Y., USA 1985).

[41] G. Kalai and D. Kleitman, A quasi-polynomial bound for the diameter of graphs of polyhedra, *Bull. Amer. Math. Soc.* **26** (1992) 315–316.

[42] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica* **4** (1984) 373–395.

[43] L.G. Khachian, Polynomial algorithms in linear programming, *Zhurnal Vichislitelnoj Matematiki i Matematischeskoi Fiziki* **20** (1980) 51–68 (in Russian), *USSR Computational Mathematics and Mathematical Physics* **20** (1980) 53–72 (English translation).

[44] E. Klafszky and T. Terlaky, Some generalizations of the criss-cross method for quadratic programming, *Math. Oper. und Stat. ser. Optimization* **24** (1992) 127–139.

[45] E. Klafszky and T. Terlaky, Some generalizations of the criss-cross method for the linear complementarity problem of oriented matroids, *Combinatorica* **9** (1989) 189–198.

[46] V. Klee and G.J. Minty, How good is the simplex algorithm? In: *Inequalities-III,* ed. O. Shisha (Academic Press, New York, 1972) 159–175.

[47] V. Klee and P. Kleinschmidt, The d–step conjecture and its relatives, *Mathematics of Operations Research* **12** (1987) 718–755.

[48] M. Kojima, N. Megiddo, T. Noma and A. Yoshise, *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*. In: Lecture Notes in Computer Science 538, (Springer Verlag 1991).

[49] C.E. Lemke, J.T., Jr. Howson, Equilibrium points of bimatrix games, *SIAM Journal* **12** (1964) 413–423.

[50] C.E. Lemke, On complementary pivot theory, In G.B. Dantzig and A.F. Veinott, editors, *Mathematics of Decision Sciences, Part 1*, 95–114, AMS, Providence, Rhode Island, 1968.

[51] T.M. Liebling, On the number of iterations of the simplex method, In R. Henn, H.P. Künzi, and H. Schubert, editors, *Methods of Operations Research XVII*, 248–264. Verlag Anton, 1972.

[52] I. Lustig, The equivalence of Dantzig's self-dual parametric algorithm for linear programs to Lemke's algorithm for linear complementarity problems applied to linear programming, *Technical Report SOL 87–4*, Department of Operations Research, Stanford University, Stanford, California, USA (1987).

[53] A. Mandel, *Topology of Oriented Matroids*, Ph.D. Thesis, (Waterloo University, Waterloo, Ontario, Canada 1982).

[54] W. Morris Jr. and M.J. Todd, Symmetry and positive definiteness in oriented matroids, *Technical Report No. 631.* Cornell University, School of Operations Research and Industrial Engineering, Itacha, N.Y. (1984).

[55] K.G. Murty, A note on a Bard type scheme for solving the complementarity problem, *Opsearch* **11 (2–3)** (1974) 123–130.

[56] K.G. Murty, *Linear and Combinatorial Programming,* (Krieger Publishing Company, Malabar, Florida, USA, 1976).

[57] K.G. Murty, Computational complexity of parametric linear programming, *Mathematical Programming* **19** (1980) 213–219.

[58] M. Namiki and T. Matsui, Some modifications of the criss-cross method, *Research Report*, Department of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan (1990).

[59] K. Paparrizos, Pivoting rules directing the simplex method through all feasible vertices of Klee-Minty examples, *Opsearch* **26(2)** (1989) 77–95.

[60] C. Roos, An exponential example for Terlaky's pivoting rule for the criss-cross simplex method, *Mathematical Programming* **46** (1990) 78–94.

[61] C. Roos, T. Terlaky and J.-Ph. Vial, *Theory and Algorithms for Linear Optimization: An Interior Point Approach* (John Wiley & Sons, New York, USA 1997).

[62] J.E. Strum, *Introduction to Linear Programming,* (Holden Day, California, USA, 1972).

[63] T. Terlaky, Egy új, véges criss-cross módszer lineáris programozási feladatok megoldására, *Alkalmazott Matematikai Lapok*, **10** (1984) 289-296. (In Hungarian, English title: A new, finite criss-cross method for solving linear programming problems.)

[64] T. Terlaky, A convergent criss-cross method, *Math. Oper. und Stat. ser. Optimization* **16(5)** (1985) 683–690.

[65] T. Terlaky, A finite criss-cross method for oriented matroids, *Journal of Combinatorial Theory (Ser. B)* **42(3)** (1987) 319–327.

[66] T. Terlaky (ed.), *Interior Point Methods of Mathematical Programming.* (Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996).

[67] T. Terlaky and S. Zhang, Pivot rules for linear programming: a survey on recent theoretical developments, *Annals of Oper. Res.* **46** (1993) 203–233.

[68] M.J. Todd, Complementarity in oriented matroids, *SIAM Journal on Algebraic and Discrete Mathematics* **5** (1984) 467–485.

[69] M.J. Todd, Linear and quadratic programming in oriented matroids, *Journal of Combinatorial Theory (Ser. B)* **39** (1985) 105–133.

[70] M.J. Todd, Polynomial expected behavior of a pivoting algorithm for linear complementarity and linear programming problems, *Mathematical Programming* **35** (1986) 173–192.

[71] A. Tucker, A note on convergence of the Ford-Fulkerson flow algorithm, *Mathematics of Operations Research* **2(2)** (1977) 143–144.

[72] H. Väliaho, A new proof of the finiteness of the criss-cross method, *Math. Oper. und Stat. ser. Optimization* **25** (1992) 391–400.

[73] H. Väliaho, P_*-matrices are just sufficient, *Linear Algebra and Its Applications* **239** (1996) 103-108.

[74] Zh. Wang, A conformal elimination free algorithm for oriented matroid programming, *Chinese Annals of Mathematics* **8(B1)** (1987).

[75] Zh. Wang, A modified version of the Edmonds-Fukuda algorithm for LP in the general form, *Asia-Pacific Journal of Operations Research* **8,1** (1991).

[76] Zh. Wang, A general deterministic pivot method for oriented matroid programming, *Chinese Annals of Mathematics B.* **13,2** (1992).

[77] Zh. Wang, T. Terlaky, A general scheme for solving linear complementarity problems in the setting of oriented matroids, (1993) In: *Combinatorics and Graph Theory*, eds. H.P. Yap, T.H. Ku, E.K. Lloyd and Zh. Wang (1993) 244–255. *Proceedings of the Spring School and International Conference on Combinatorics: SSIC'92, China*, World Scientific, Singapore.

[78] S.J. Wright, *Primal-Dual Interior Point Methods.* SIAM Publications, Philadelphia, USA (1996).

[79] N. Zadeh, What is the worst case behavior of the simplex algorithm? *Technical Report No. 27*, Department of Operations Research, Stanford University, Stanford, California, USA (1980).

[80] S. Zhang, On anti-cycling pivoting rules for the simplex method, *Operations Research Letters* **10** (1991) 189–192.

[81] S. Zhang, New variants of finite criss-cross pivot algorithms for linear programming, *Technical Report, Econometric Institute, Erasmus University Rotterdam* **9707/A** (1997).

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

[83] S. Zionts, Some empirical test of the criss-cross method, *Management Science* **19** (1972) 406-410.