DUAL FACE ALGORITHM USING GAUSS-JORDAN ELIMINATION FOR LINEAR PROGRAMMING

PING-QI PAN

Abstract. The dual face algorithm uses Cholesky factorization, as would be not very suitable for sparse computations. The purpose of this paper is to present a dual face algorithm using Gauss-Jordan elimination for solving bounded-variable LP problems.

1. Introduction

The simplex method ([1]–[3]) starts from an initial feasible basis, and goes from basis to basis, until reaching an optimal basis. Some variants of the simplex method are more flexible, not confining to feasible bases. For example, criss-cross algorithms [19, 16, 17, 18, 4], switch between primal and dual simplex steps. Other approaches [5]–[8] even no longer produce standard bases.

Reflects further efforts along this line, the deficient basis ([9]-[14]) is defined as a submatrix from the coefficient matrix, whose range space includes the right hand side of the system (in the standard problem). Seeming to be a "fly in the ointment", however, such a concept does not accommodate the case when the condition is not satisfied, even if related computational results are very favorable against the simplex method.

Independently developed, the concept of dual face [15](pp.595) turns out to be a good answer in this respect. The dual face algorithm may be regarded as an extension of the dual simplex method, allowing itself to start from scratch by selecting columns, one by one, from the coefficient matrix. Initiated from any feasible point, it proceeds from dual face to dual face until reaching an optimal dual face, together with an optimal dual solution on it. In computational experiments with a set of 26 small Netlib standard problems, the dual face algorithm outperformed the simplex algorithm with overall time ratio as high as 10.04 — an incredible outcome indeed! This should not be surprising, as most systems handled in its solution process are far smaller than conventional ones; and the former solves only a single triangular system in each iteration, compared with the latter solving four such systems.

Nevertheless, the dual face algorithm uses Cholesky factorization, as would be not very suitable for sparse computations, compared with Gauss elimination. From a practicable point of view, in addition, there is a need for generalizing the dual face algorithm to handle the bounded-variable LP problem below:

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b, \quad l \leq x \leq u, \end{array} \tag{1.1}$$

1991 *Mathematics Subject Classification.* Primary 90C05; Secondary 65K05.

Key words and phrases. linear programming, bounded-variable, dual face, dual optimal face, Gauss-Jordan elimination .

Projects 10871043 and 70971136 supported by National Natural Science Foundation of China.

where $A \in \mathcal{R}^{m\times n}$, $m < n$, $Ax = b$ is consistent, and l and u are finite lower and upper bound vectors. Note that A is not necessarily of full row rank, in contrast to the conventional assumption in simplex contexts.

The purpose of this paper is to present a generalized dual face algorithm. For simplicity of exposition, the algorithm is described using Gauss-Jordan elimination. It is organized as follows. Firstly, section 2 derives the search direction via local duality. Section 3 addresses updating of dual feasible solution, and the associated pivot operations. Section 4 highlights the optimality test and associated pivot operations when failing with the test. Section 5 gives the the generalized algorithm in tableau form, and Section 6 converts it to a practicable revised version.

2. Search Direction

For succinctness, we put (1.1) in the following tableau

(2.1)

x^T	f	RHS
A		b
c^T	-1	

For any integer $1 \le k \le m$, let

$$B = \{j_1, \cdots, j_k\} \qquad \text{and} \qquad N = \{1, \cdots, n\}\backslash B.$$

be a partition to the index set. Without confusion, such notations for index sets will also be used to denote matrices consisting of corresponding columns. Partition the row index set to

$$R = \{i_1, \cdots, i_k\}, \qquad \text{and} \qquad R' = \{1, \cdots, m\}\backslash R.$$

Denote by $B_R \in \mathcal{R}^{k\times k}$ the submatrix indexed by R and B, and do similarly with others. Assume that B_R is nonsingular. Then tableau (2.1) can be put into the following form:

x_B^T	x_N^T	f	RHS
B_R	N_R		b_R
$B_{R'}$	$N_{R'}$		$b_{R'}$
c_B^T	c_N^T	-1	

Assume that Gauss elimination transforms the preceding tableau to

(2.2)

x_B^{T}	x_N^{T}	f	$\bar{b}$
U	$\bar{N}_R$		$\bar{b}_R$
	$\bar{N}_{R'}$		$\bar{b}_{R'}$
	$\bar{z}_N^T$	-1	μ

where $U \in \mathcal{R}^{k\times k}$ is a nonsingular upper triangular matrix.

The dual program of the preceding tableau (program) is

$$\begin{array}{lllllll} \max & \bar{b}_R^T y_R + \bar{b}_{R'}^T y_{R'} + l^T v + u^T w & & & & & \\ \text{s.t.} & y_R & & +v_B & +w_B & = & 0 \\ & \bar{N}_R^T y_R & +\bar{N}_{R'}^T y_{R'} & +v_N & +w_N & = & \bar{z}_N \end{array} \qquad v \ge 0, \quad w \le 0. \tag{2.3}$$

Let (Γ, Π) be a partition to N such that

$$\Gamma = \{\ j \in N \mid \bar{z}_j \ge 0\}, \quad \Pi = \{\ j \in N \mid \bar{z}_j < 0\}. \tag{2.4}$$

Consider dual solution $(\bar{y}, \bar{v}, \bar{w})$ such that

$$\begin{array}{rclcl rcl} \bar{y}_R & = & 0, & & & \bar{v}_B, \bar{w}_B & = & 0, \\ \bar{v}_\Gamma & = & \bar{z}_\Gamma - \bar{\Gamma}_{R'}^{\mathrm{T}} \bar{y}_{R'} & \geq & 0, & \bar{w}_\Gamma & = & 0, \\ \bar{w}_\Pi & = & \bar{z}_\Pi - \bar{\Pi}_{R'}^{\mathrm{T}} \bar{y}_{R'} & \leq & 0, & \bar{v}_\Pi & = & 0. \end{array} \tag{2.5}$$

It is verified that the preceding is a feasible solution to the dual problem (2.3).

The so-called "local dual program" (Pan, 2014) associated with (2.5) is

$$\begin{array}{lllll} \max & \bar{b}_R^T y_R + \bar{b}_{R'}^T y_{R'} + l_\Gamma^T v_\Gamma + u_\Pi^T w_\Pi & & & \\ & y_R & = & 0, & \\ \text{s.t.} & \quad \bar{\Gamma}_{R'}^{\mathrm{T}} y_{R'} \quad + v_\Gamma & = & \bar{z}_\Gamma, & v_\Gamma \geq 0, w_\Pi \leq 0. \\ & \quad \bar{\Pi}_{R'}^{\mathrm{T}} y_{R'} \qquad + w_\Pi & = & \bar{z}_\Pi, & \end{array}$$

Variables y_R and v_Γ, w_Π in the objective function can be eliminated via the constraints. Introduce notation

$$z_\Gamma = v_\Gamma, \qquad z_\Pi = w_\Pi.$$

Dropping $y_R = 0$ aside, we thereby transform the local dual program to the following compact form:

$$\begin{array}{ll} \max & (\bar{b}_{R'} - \bar{\Gamma}_{R'} l_\Gamma - \bar{\Pi}_{R'} u_\Pi)^T y_{R'} \triangleq \tilde{b}_{R'}^T y_{R'} \\ \text{s.t.} & N_{R'}^{\mathrm{T}} y_{R'} + z_N = \bar{z}_N, \qquad z_\Gamma \geq 0, z_\Pi \leq 0. \end{array} \tag{2.6}$$

It is clear that $(\bar{y}_{R'} = 0, \bar{z}_N)$ is a feasible solution to the preceding program.

Define

$$\bar{x}_\Gamma = l_\Gamma, \qquad \bar{x}_\Pi = u_\Pi. \tag{2.7}$$

We adapt tableau (2.2) by replacing the right-hand side $\bar{b}$ by $\tilde{b} \triangleq \bar{b} - \bar{\Gamma}\bar{x}_\Gamma - \bar{\Pi}\bar{x}_\Pi$ and dropping the f column (as will not be changed in the sequel), i.e.,

(2.8)

x_B^{T}	x_N^{T}	$\tilde{b}$
U	$\bar{N}_R$	$\tilde{b}_R$
	$\bar{N}_{R'}$	$\tilde{b}_{R'}$
	$\bar{z}_N^T$	$\tilde{\mu}$

where $\tilde{\mu} = \mu - \bar{z}_\Gamma^T \bar{x}_\Gamma - \bar{z}_\Pi^T \bar{x}_\Pi$. The preceding, referred to as *(dual) face tableau* (for the reason which will be clear later), corresponds to dual feasible solution $(\bar{z}_B = 0, \bar{z}_N)$ (with $\bar{y} = 0$). The latter exhibits complementarity slackness with $\bar{x}$ (with $\bar{x}_B = U^{-1}\tilde{b}_R$), even if $\bar{x}$ is not a primal solution in strict sense when $\tilde{b}_{R'} \neq 0$.

Taking the objective gradient $\tilde{b}_{R'}$ as search direction in $y_{R'}$ space, we have the following set of search vectors:

$$\Delta y_{R'} = \tilde{b}_{R'}, \qquad \Delta z_N = -\bar{N}_{R'}^T \tilde{b}_{R'}. \tag{2.9}$$

It is seen that $\Delta y_{R'}$ vanishes if $\tilde{b}_{R'} = 0$ or $R' = \emptyset$ $(k = m)$. On the other hand, $\Delta y_{R'} \neq 0$ if $k < m$ and $\tilde{b}_{R'} \neq 0$, since $Ax = b$ is consistent. In addition, $\Delta y_{R'}$ is readily available as $\tilde{b}_{R'}$ can be read directly from the right-hand side of face tableau (2.8). Even Δz_N can be calculated easily. In fact, $\Delta z_N^T = -\tilde{b}_{R'}^T \bar{N}_{R'}$ is the negative linear combination of rows of $\bar{N}_{R'}$ (with $\tilde{b}_{R'}$ as combination coefficients).

3. Updating Dual Solution

Assume now that $\tilde{b}_{R'} \neq 0$, and $(\Delta y_{R'}, \Delta z_N)$ has been calculated by (2.9). Then, a solution results from the the following line search scheme:

$$\hat{y}_{R'} = \bar{y}_{R'} + \beta \Delta y_{R'}, \tag{3.1}$$

$$\hat{z}_N = \bar{z}_N + \beta \Delta z_N, \tag{3.2}$$

It is clear that the preceding fulfils the equality constraint of (2.6) for any $\beta \geq 0$. Further, satisfying the other constraints ($z_\Gamma \geq 0$ and $z_\Pi \leq 0$) gives the largest possible stepsize β and the according index q, i.e.,

$$\beta = -\bar{z}_q/\Delta z_q = \min \left\{ -\bar{z}_j/\Delta z_j \;\middle|\; \begin{array}{ll} \Delta z_j < 0, & j \in \Gamma \\ \Delta z_j > 0. & j \in \Pi \end{array} \right\} \tag{3.3}$$

Therefore, formulas (3.1) and (3.2) give a new dual feasible solution to (2.6). That is to say, $\hat{z}_N$ can be used to update the bottom line of the face tableau (2.8)(see Proposition 4.7.1 in Pan, 2014).

Note that the stepsize β would vanish, yielding a solution just the same as the old, if there is some component of $\bar{z}_N$ being equal to zero, a case said to be *dual degenerate*.

We have the following Lemma.

Lemma 3.1. *Assume that* $\Delta y_{R'} \neq 0$.

(i) If $\Delta z_\Gamma \geq 0$ *and* $\Delta z_\Pi \leq 0$*, problem (1.1) is upper unbounded; else,*

(ii) $(\hat{y}_{R'}, \hat{z}_N)$ *is a boundary point of the feasible region, with the dual objective value not decreasing, or even strictly increasing if* $\bar{z}_N$ *is nondegenerate.*

Proof. (i) Since $(\bar{y}_{R'}, \bar{z}_N)$ is dual feasible, the new point $(\hat{y}, \hat{z}_N)$ is a well-defined dual feasible solution for all $\beta \geq 0$. Noting $\Delta y_R = 0$, we obtain from the first formula of (2.9) that

$$\tilde{b}^T \Delta y = \tilde{b}_{R'}^T \tilde{b}_{R'} \geq 0,$$

which together with $\Delta y_{R'} \neq 0$ gives

$$\tilde{b}^T \Delta y > 0.$$

Therefore, it holds that

$$\tilde{b}^T \hat{y} = \tilde{b}^T \bar{y} + \beta \tilde{b}^T \Delta y \geq \tilde{b}^T \bar{y}, \tag{3.4}$$

which implies that the objective value tends to $+\infty$, as so does β.

(ii) It is seen from (3.4) that the associated objective value does not decrease, and strictly increases in case of nondegeneracy ($\beta > 0$). It is also clear that $\hat{z}_q = 0$, indicating that the new iterate is on the boundary. □

Since $\hat{z}_q = 0$ blocks any further increase of the stepsize, index q should be entered to B. To determine leaving index j_p, we select row index p such that

$$p \in \arg\max_{i \in R'} |\bar{a}_{i\,q}|, \tag{3.5}$$

where $\bar{a}_{i\,q}, i \in R'$ are entries of the q-index column of $\bar{N}_{R'}$ in tableau (2.8).

4. Optimality Test

If $\tilde{b}_{R'} = 0$, the associated search direction vectors are meaningless. Nevertheless, this offers a chance to test for optimality.

Now, $\bar{x}$ defined by (2.7) together with $\bar{x}_B = U^{-1}\tilde{b}_R$ is a primal solution. Moreover, the solution is feasible if it holds that

$$l_B \leq \bar{x}_B \leq u_B. \tag{4.1}$$

Since the preceding exhibits complementarity slackness with $(\bar{z}_B = 0, \bar{z}_N)$, in fact, these are a pair of primal and dual optimal solutions.

For more precise, introduce the following quantities:

$$\rho_i = \begin{cases} l_{j_i} - \bar{x}_{j_i}, & \text{if } \bar{x}_{j_i} < l_{j_i}, \\ u_{j_i} - \bar{x}_{j_i}, & \text{if } \bar{x}_{j_i} > u_{j_i}, \\ 0, & \text{if } l_{j_i} \leq \bar{x}_{j_i} \leq u_{j_i}, \end{cases} \qquad i = 1, \cdots, k. \tag{4.2}$$

Determine an index j_s such that

$$s \in \arg\max\{|\rho_i| \mid i = 1, \cdots, k\}. \tag{4.3}$$

If $\rho_s = 0$, implying (4.1), the optimality is achieved, and we are done. Assume that $\rho_s \neq 0$. The according index j_s is dropped from B.

Then, B is updated by

$$\hat{B} = B \backslash \{j_s\}. \tag{4.4}$$

If $\rho_s > 0$, implying that $\bar{x}_p$ violates the lower bound, j_s is moved to Γ with Π remaining unchanged, while if $\rho_s < 0$, j_s is moved to Π with Γ unchanged.

Accordingly, the leaving component of $\bar{x}$ is updated by

$$\bar{x}_{j_s} = \bar{x}_{j_s} + \rho_s,$$

which is on the lower or upper bound

5. The Tableau Algorithm

It is seen that the face tableau of form (2.8) includes all information needed to carry out operations. It can be updated iteration by iteration There are two types of iterations:

(i) Rank-increasing: $\tilde{b}_{R'} \neq 0$. The bottom line of the tableau is updated. A column index and a row index are respectively added to B and R, and Therefore, the column rank of the face matrix increases by one.

(ii) Rank-decreasing: $\tilde{b}_{R'} = 0$. If $l_R \leq \tilde{b}_R \leq u_R$, optimality has been achieved. Otherwise, a column index and a row index are respectively taken off from B and R. the column rank of the face matrix decreases by one.

In each iteration, the tableau is updated by elementary transformations, as sets B, Γ, Π, R, R' change. In addition, the right-hand side $\tilde{b}$ must be adapted accordingly.

The overall steps may be summarized in tableau form as follows.

Algorithm 1.(Generalized dual face algorithm using Gauss-Jordan elimination: tableau form) Initial face tableau of form (2.8) with $1 \leq k \leq m$; B, Π, Γ, R, R' and $\tilde{b}$; dual feasible solution $(\bar{y}, \bar{z})$ associated with $\bar{x}_N$; This algorithm solves the bounded-variable problem (1.1).

1. Go to step 10 if $R' = \emptyset$ or $\tilde{b}_{R'} = 0$.

2. Compute $\Delta z_N = -\bar{N}(R')^T \tilde{b}_{R'}$, $\Delta z_B = 0$.
3. Stop if $\Delta z_\Gamma \geq 0$ and $\Delta z_\Pi \leq 0$.
4. Determine index q and stepsize β by (3.3).
5. If $\beta \neq 0$, add β times of Δz to the bottom line.
6. Determine $p = \arg\max\{|\bar{a}_{i\,q}| \mid i \in R'\}$.
7. Convert $\bar{a}_{p\,q}$ to 1, and eliminate the other nonzeros in the column.
8. Update $\tilde{b}_p = \tilde{b}_p + \bar{x}_q$; move p from R' to R, q from N to B;
9. Set $k = k + 1$, and go to step 1 if $k < m$.
10. Set $\bar{x}_B = \tilde{b}_R$.
11. Determine row index s by (4.3), where ρ_i is computed by (4.2).
12. Stop if $\rho_s = 0$.
13. Update $\bar{x}_{i_s} = \bar{x}_{i_s} + \rho_s$.
14. Update $\tilde{b}_{i_s} = -\rho_s$; move i_s from R to R';
 if $\rho_s < 0$, move j_s from B to Γ, else to Π.
15. Compute $\Delta z_N = -\text{sign}(\tilde{b}_{i_s})\bar{N}_(i_s)^T$, $\Delta z_B = 0$.
16. Set $k = k - 1$, and go to step 4.

Theorem 5.1. *Under the nondegeneracy assumption, Algorithm 1 terminates either at*
(i) step 3, detecting lower unboundedness of the problem; or at
(ii)step 12, generating a pair of primal and dual optimal solutions.

6. The Revised Algorithm

To be practicable tableau Algorithm 1 is revised in this section.

Assume that B_R^{-1} is available. Then tableau (2.8) can be put to the revised form equivalently, i.e.,

(6.1)

$$\begin{array}{cc|c} x_B^{\mathrm{T}} & x_N^{\mathrm{T}} & \tilde{b} \\ \hline I & B_R^{-1}N_R & \tilde{b}_R \\ & N_{R'} - B_{R'}B_R^{-1}N_R & \tilde{b}_{R'} \\ \hline & \bar{z}_N^T & \tilde{\mu} \end{array},$$

where $\tilde{b}, \bar{z}_N^T$ and $\tilde{\mu}$ are updated in each iteration. From the preceding, the search direction in z_N space comes, i.e.,

$$\Delta z_N = N_R^T v - N_{R'}^T \tilde{b}_{R'}, \qquad B_R^T v = B_{R'}^T \tilde{b}_{R'}.$$

(i) Rank-increasing iteration: $R' \neq \emptyset$ and $\tilde{b}_{R'} \neq 0$. Then $\Delta z_N \neq 0$. Assume that pivot column index q and row index $p \in R'$ were determined by (3.3) and (3.5), respectively.

The effect of the elementary transformations in step 7 of Algorithm 1 amounts to premultiplying the matrix of form

$$E_p = \begin{pmatrix} 1 & & -\bar{a}_{1,\,q}/\bar{a}_{p,\,q} & & \\ & \ddots & \vdots & & \\ & & -\bar{a}_{p-1,\,q}/\bar{a}_{p,\,q} & & \\ & & 1/\bar{a}_{p,\,q} & & \\ & & -\bar{a}_{p+1,\,q}/\bar{a}_{p,\,q} & & \\ & & \vdots & \ddots & \\ & & -\bar{a}_{m,\,q}/\bar{a}_{p,\,q} & & 1 \end{pmatrix}$$

Therefore, the right-hand side should be updated by

$$\widehat{b} = E_p\tilde{b}, \tag{6.2}$$

On the other hand, the matrix B_R is updated by

$$\hat{B}_{\hat{R}} = \begin{array}{cc} \left(\begin{array}{c|c} B_R & a_{R,\,q} \\ \hline e_p^{\mathrm{T}}B & a_{p\,q} \end{array}\right) & \begin{array}{c} k \\ 1 \end{array} \\ \begin{array}{cc} k & \quad 1 \end{array} & \end{array} \tag{6.3}$$

where $e_p^{\mathrm{T}}B$ dentes the pth row of B, $a_{p\,q}$ denotes the pth component of a_q, and $a_{R,\,q}$ corresponds to face components of a_q. It is easy to verify that the inverse is of the following form

$$\hat{B}_{\hat{R}}^{-1} = \begin{array}{cc} \left(\begin{array}{c|c} U & v \\ \hline d^{\mathrm{T}} & \tau \end{array}\right) & \begin{array}{c} k \\ 1 \end{array} \\ \begin{array}{cc} k & \quad 1 \end{array} & \end{array} \tag{6.4}$$

where

$$\begin{aligned} h^{\mathrm{T}} &= e_p^{\mathrm{T}}B, \\ \tau &= 1/(a_{p\,q} - h^{\mathrm{T}}\bar{a}_{R,\,q}), \\ v &= -\tau\bar{a}_{R,\,q}, \\ d^{\mathrm{T}} &= -\tau h^{\mathrm{T}}B_R^{-1}, \\ U &= (I - vh^{\mathrm{T}})B_R^{-1}. \end{aligned}$$

(ii) Rank-remaining iteration$R' = \emptyset$ or $\tilde{b}_{R'} = 0$.

Assume that s was determined by (4.3). Then B_R^{-1} is updated simply by dropping its sth row and column.

Algorithm 2. (Generalized dual face algorithm using Gauss-Jordan elimination) Initial: $1 \le k \le m$; B, Π, Γ, R, R', B_R^{-1} and $\tilde{b}$; dual feasible solution $(\bar{y}, \bar{z})$ associated with $\bar{x}_N$; This algorithm solves the bounded-variable problem (1.1).

1. Go to step 11 if $R' = \emptyset$ or $\tilde{b}_{R'} = 0$.
2. Compute $\Delta z_N = N_R^T v - N_{R'}^T \tilde{b}_{R'}$, where $B_R^T v = B_{R'}^T \tilde{b}_{R'}$.
3. Stop if $\Delta z_\Gamma \ge 0$ and $\Delta z_\Pi \le 0$ (unbounded problem).
4. Determine index q and stepsize β by (3.3).
5. If $\beta \ne 0$, updated $\bar{z}_N$ by (3.2).
6. Compute $\bar{a}_{R'\,q} = a_{R'\,q} - B_{R'}w$, where $w = B_R^{-1}a_{R\,q}$.
7. Determine $p = \arg\max\{|\bar{a}_{i\,q}| \mid i \in R'\}, i_s = p$.
8. Update $\tilde{b}$ by (6.2) and $\tilde{b}_p = \tilde{b}_p + \bar{x}_q$;.
9. Move p from R' to R, q from N to B;
10. Set $k = k + 1$, and go to step 1 if $k < m$.
11. Set $\bar{x}_B = \tilde{b}_R$.
12. Determine row index s by (4.3), where ρ_i is computed by (4.2).
13. Stop if $\rho_s = 0$ (optimality achieved).
14. Update $\bar{x}_{i_s} = \bar{x}_{i_s} + \rho_s$.
15. Update $\tilde{b}_{i_s} = -\rho_s$; move i_s from R to R';
 if $\rho_s < 0$, move j_s from B to Γ, else to Π.
16. Compute $\Delta z_N = -\mathrm{sign}(\tilde{b}_{i_s})\bar{N}_(i_s)^T$.
17. Set $k = k - 1$, and go to step 4.

References

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

[2] G.B. Dantzig, *Programming of interdependent activities, mathematical model*, Econometrica, 17(1949), No.3/4 , 200-211.

[3] G.B. Dantzig, *Maximization of a linear function of variables subject to linear inequalities*, in *Activity Analysis of Production and Allocation* (T.C.Koopmans,ed.), Wiley, New York, 339-347, 1951.

[4] Wei Li, *Practical Criss-Cross Method for Linear Programming*, Advances in Neural Networks - ISNN 2010, Lecture Notes in Computer Science, 6063 (2010), 223-229.

[5] P.-Q. Pan, *Practical finite pivoting rules for the simplex method*, OR Spektrum, 12 (1990), 219–225.

[6] P.-Q. Pan, *A simplex-like method with bisection for linear programming*, Optimization, 22 (1991), No. 5, 717–743.

[7] P.-Q. Pan, *A modified bisection simplex method for linear programming*, J. of Computational Mathematics, 14 (1996), No. 3, 249–255.

[8] P.-Q. Pan, *The most-obtuse-angle row pivot rule for achieving dual feasibility: a computational study*, European J. of Operational Research, 101 (1997), 167–176.

[9] P.-Q. Pan, *A dual projective simplex method for linear programming*, Computers and Mathematics with Applications, 35 (1998), No. 6, 119–135.

[10] P.-Q. Pan, *A basis-deficiency-allowing variation of the simplex method*, Computers and Mathematics with Applications, 36 (1998), No. 3, 33–53.

[11] P.-Q. Pan, *A projective simplex method for linear programming*, Linear Algebra and Its Applications, 292 (1999), 99–125.

[12] P.-Q. Pan, *A projective simplex algorithm using LU factorization*, Computers and Mathematics with Applications, 39 (2000), 187–208.

[13] P.-Q. Pan, *A dual projective pivot algorithm for linear programming*, Comput. Optim. Appl., 29 (2004), 333–344.

[14] P.-Q. Pan, *A revised dual projective pivot algorithm for linear programming*, SIAM J. on Optimization, 16 (2005), No.1, 49–68.

[15] P.-Q. Pan, *Linear Programming Computation*, Springer, Berlin Heidelberg, Germany, 2014.

[16] T. Terlaky, *A covergent criss-cross method*, Math.,Oper. Undsta. Ser.Optimization, 16 (1985), No. 5, pp. 683–690.

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

[18] Yan H-Y, Pan P-Q (2009) Most-obtuse-angle criss-cross algorithm for linear programming (in Chinese). Numer Math J Chin Univ 31:209C215.

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

Department of Mathematics, Southeast University, Nanjing, 210096, China
E-mail address: panpq@seu.edu.cn