A NEW DUAL FACE ALGORITHM USING LU FACTORIZATION FOR LINEAR PROGRAMMING

PING-QI PAN//DEPARTMENT OF MATHEMATICS// SOUTHEAST UNIVERSITY// NANJING 210096, CHIN

Abstract. The dual face algorithm for linear programming (LP) was proposed by the author in 2014. Using QR factorization, it proceeds from dual face to dual face, until reaching an optimal dual face along with dual and primal optimal solutions, unless detecting infeasibility of the problem. On the other hand, a variant of the algorithm using LU factorization was presented, but it is for specialized LP problems only. To break the barriers, this paper proposes a new dual face algorithm using LU factorization for standard LP problems. Surprisingly enough, it turns out that this algorithm and the face algorithm [17] just proposed are a pigeon pair. Like the latter, the former also has salient features of its own, and is even more attractive.

1. Introduction

Consider the linear programming (LP) problem in the following standard form:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \quad x \geq 0, \end{aligned} \tag{1.1}$$

where $A \in R^{m\times n}$ $(m < n)$, $b \in R^m$, $c \in R^n$. No assumption is made on the rank of A, except for the equality constraints being consistent.

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

Key words and phrases. linear programming, pseudo-basis, deficient-basis, dual face, LU factorization .

Since established by George B. Dantzig in 1947, the simplex algorithm [3, 4] for solving LP problems has been widely used in practice for more than 70 years. The algorithm is usually fast, if implemented wisely. Playing a central role in the simplex methodology, nevertheless, the *basis* causes degeneracy, degrading performance of the simplex algorithm in solving large difficult LP problems.

To cope this problem, a so-called " perturbation approach" [2] added a small perturbation term to the right-hand side to affect column selection under Dantzig's original rule. On the other hand, efforts were also made to design finite rules, such as those proposed in [1, 6]. Although there is no risk of cycling in theory, however, these rules are far less efficient than Dantzig's rule, in practice.

More recently, introduced was the so-called *deficient-basis* [8]. As a result, involved systems are smaller, and the related solution has fewer zero components, relieving degeneracy to some extent. Several variants of the deficient-basis algorithm were proposed with remarkable computational results [9]-[14]. Nevertheless, it is a bit disappointing that columns of a deficient-basis may increase in the solution process.

Primal and dual face algorithms were first published in 2014 [15] with preliminary but significant computational results. Motivated by degeneracy, both algorithms were based on QR factorization. A variant of the latter was developed using LU factorization, but it is designed for problems in a specialized form only (section 23.3, [15]).

To break the barriers, this paper proposes a new dual face algorithm using LU factorization for solving standard LP problems. Surprisingly enough, it turns out that this algorithm and the face algorithm just proposed [17] are a pigeon pair. It is interesting to compare the two algorithms. Like the later, the former also has salient features of its own, and is even more attractive.

In Section 2, the previous work is reviewed first. In Section 3, a fresh new dual search direction is developed using LU factorization. In section 4, discussed is update of dual feasible solution. Section 5 is devoted to pivoting operations. We discuss optimality condition in section 6. Then, we formulate the new dual face algorithm in tableau form in Section 7 and in revised form in section 8. Finally, we make remarks in section 9.

In this paper, a_j denotes the j-indexed column of matrix A, and B_R the matrix with columns indexed by B and rows indexed by R, and so on. For simplicity, the same notation for every matrix is also used for the corresponding index set of their columns or rows. As usual, I_j denotes $j \times j$ unit matrix, and e_j unit vector with its j-th component 1.

2. Previous work

In this section, we review the original dual face algorithm and a variant of it, focusing on their search directions, in particular.

To begin with, we introduce the following concept [12].

Definition 2.1. *Pseudo-basis is a submatrix of A with full column rank* [1].

The following concept appeared for the first in [8].

Definition 2.2. *Deficient-basis is a pseudo-basis of A, whose column space includes b.*

If $k = m$, the deficient-basis is a normal basis.

Now turn to the dual problem of (1.1), i.e.,

$$\begin{aligned} &\max \quad b^T y \\ &\text{s.t.} \quad A^T y + z = c, \quad z \geq 0. \end{aligned} \tag{2.1}$$

[1]Empty submatrix is viewed as pseudo-basis with rank 0.

Given partition $A = B \cup N$, where B is $m \times k$ pseudo-basis with $0 \leq k \leq m$.

Definition 2.3. *Dual face is a dual feasible point (solution) set, defined as*

$$D_N = \{(y, z_N) \in \mathcal{R}^m \times \mathcal{R}^{n-k} \mid B^T y = c_B,\ N^{\mathrm{T}} y + z_N = c_N,\ z_B = 0,\ z_N \geq 0\}, \tag{2.2}$$

N and B are termed *dual face matrix* and *dual nonface matrix*, respectively. As $z_B = 0$ is fixed, B is said *inactive*.

The dual face D_N is an $(m-k)-$dimensional set. In particular, if $k = m$, D_N is 0-dimensional, a dual vertex; if $k = 0$, it is m-dimensional, the dual feasible region itself. Usually, $0 < k < m$ is assumed. Any solution in D_N is called *dual face point (solution).*

Definition 2.4. *Dual face is a dual level face if the dual objective value is constant over it. A dual level face is optimal if the constant is equal to the dual optimal value.*

The original dual face algorithm is based on the following least squares problem:

$$\min_{x_B \in \mathcal{R}^k} \|b - Bx_B\|_2. \tag{2.3}$$

The unique solution to the preceding is known as

$$\bar{x}_B = (B^T B)^{-1} B^T b, \tag{2.4}$$

which can be obtained by solving the so-called *normal system* below:

$$(B^T B)\bar{x}_B = B^T b. \tag{2.5}$$

The residual of (2.3) at solution $\bar{x}_B$ is a key quantity, that is,

$$\Delta y = b - B\bar{x}_B. \tag{2.6}$$

It is clear that Δy vanishes if and only if B is a deficient-basis.

In the other case, it can be verified that

$$B^T\Delta y = 0, \qquad b^T\Delta y > 0, \tag{2.7}$$

which means that nonzero Δy is in the null of B^T, and is an uphill with respect to the dual objective. In fact, Δy is just the orthogonal projection of b onto the null of B^T. As the closest to the dual objective gradient b in the null of B^T, it is a favorite search direction in y-space.

The associated search direction in z-space is obtained from the equality constraint in (2.1), that is,

$$\Delta z_B = 0, \qquad \Delta z_N = -N^{\mathrm{T}}\Delta y. \tag{2.8}$$

To obtain Δy, the original dual face algorithm (section 23.1, [15]) computes $\bar{x}_B$ by handling normal system (2.5) via Cholesky factorization of B^TB. The wanted Cholesky factor is yielded from QR factorization of B, and updated iteration by iteration subsequently. Computational results are preliminary but significant. A related dense code outperformed RSA, a simplex code, with time ratio of 10.04 on a set of 25 smallest standard Netlib problems.

On the other hand, a variant using LU factorization is proposed for specialized LP problems (section 23.3, [15]). In the following, we outline this approach in the context of the standard LP problem instead.

Without loss of generality, assume that B consists of the first k columns, and R the first k rows, such that $k\times k$ submatrix B_R has full rank.

Using notation

$$N = A\backslash B, \qquad R_c = \{1,\cdots,m\}\backslash R,$$

we put problem (1.1) into the following tableau:

(2.9)

x_B^{T}	x_N^{T}	f	RHS
B_R	N_R		b_R
B_{R_c}	N_{R_c}		b_{R_c}
c_B^{T}	c_N^{T}	-1	

.

Carrying out Gauss-Jordan elimination to the preceding, we convert B_R into unit matrix, and eliminate all entries below it, yielding the following tableau:

(2.10)

x_B^{T}	x_N^{T}	f	RHS
I_k	$\bar{N}_R$		$\bar{b}_R$
	$\bar{N}_{R_c}$		$\bar{b}_{R_c}$
	$\bar{c}_N^T$	-1	$-\bar{f}$

where $\bar{c}_N$ is assumed nonnegative.

Tableau (2.10) corresponds to the following dual subproblem:

(2.11)

$$\begin{array}{ll} \max & \bar{b}_R^T y_R + \bar{b}_{R_c}^T y_{R_c} \\ \text{s.t.} & \begin{pmatrix} I_k & \\ \bar{N}_R^T & \bar{N}_{R_c}^T \end{pmatrix} \begin{pmatrix} y_R \\ y_{R_c} \end{pmatrix} + \begin{pmatrix} z_B \\ z_N \end{pmatrix} = \begin{pmatrix} 0 \\ \bar{c}_N \end{pmatrix}, \qquad z_B = 0,\ z_N \geq 0. \end{array}$$

For the preceding, the counterpart of least squares problem (2.3) is much simpler, that is,

$$\min_{x_B \in \mathcal{R}^k} \left\| \begin{pmatrix} \bar{b}_R \\ \bar{b}_{R_c} \end{pmatrix} - \begin{pmatrix} I_k \\ 0 \end{pmatrix} x_B \right\|^2, \tag{2.12}$$

which exhibits its unique solution $\bar{x}_B = \bar{b}_R$ with residual

$$\Delta y = \begin{pmatrix} 0 \\ \bar{b}_{R_c} \end{pmatrix} \tag{2.13}$$

If $\bar{b}_{R_c} \neq 0$, Δy is eligible to be a search direction in y-space. The matching search direction in z-space is then

$$\Delta z_B = 0, \quad \Delta z_N = -\bar{N}_{R_c}^{\mathrm{T}} \bar{b}_{R_c}. \tag{2.14}$$

Tableau (2.9) can also be handled by block Gauss-Jordan elimination. Premultiplying the first row by B_R^{-1}, adding $-B_{R_c}$ times of the first row to the second row, and adding $-c_B^T$ times of the first row to the bottom row, we convert (2.9) to the following tableau:

(2.15)

x_B^{T}	x_N^{T}	f	RHS
I_k	$B_R^{-1}N_R$		$B_R^{-1}b_R$
	$N_{R_c} - B_{R_c}B_R^{-1}N_R$		$b_{R_c} - B_{R_c}B_R^{-1}b_R$
	$c_N^T - c_B^T B_R^{-1}N_R$	-1	$-c_B^T B_R^{-1}b_R$

Since the preceding is equivalent to (2.10), their corresponding entries are equal, e.g.,

$$\bar{c}_N = c_N - N_R^T B_R^{-T} c_B, \qquad \bar{N}_{R_c} = N_{R_c} - B_{R_c} B_R^{-1} N_R. \tag{2.16}$$

From the equality constraint of (2.11), it is seen that $y_R = 0$ is actually fixed since so is $z_B = 0$. Dropping such an inactive part, subproblem (2.11) becomes

$$\begin{array}{ll} \max & \bar{b}_{R_c}^T y_{R_c} \\ \text{s.t.} & \bar{N}_{R_c}^T y_{R_c} + z_N = \bar{c}_N, \qquad z_N \geq 0. \end{array} \tag{2.17}$$

Designating the feasible region of the preceding by D'_N, we have the following result.

Lemma 2.1. *If* $\{y_{R_c}, z_N \geq 0\}$ *is in* D'_N*, then* $\{z_B = 0,\ y_R = B_R^{-T}(c_B - B_{R_c}^T y_{R_c}),\ y_{R_c},\ z_N \geq 0\}$ *is in* D_N*, defined by (2.2), and vice versa.*

Proof.

$$\bar{c}_N = c_N - N_R^T B_R^{-T} c_B, \qquad \bar{N}_{R_c} = N_{R_c} - B_{R_c} B_R^{-1} N_R. \tag{2.18}$$

Assume that the former point is in D'_N. From definition of D'_N and (2.18), it follows that

$$\begin{aligned} z_N &= \bar{c}_N - \bar{N}_{R_c}^T y_{R_c} \\ &= c_N - N_R^T B_R^{-T} c_B - (N_{R_c}^T - N_R^T B_R^{-T} B_{R_c}^T) y_{R_c} \\ &= c_N - N_{R_c}^T y_{R_c} - N_R^T B_R^{-T}(c_B - B_{R_c}^T y_{R_c}), \end{aligned} \tag{2.19}$$

Substituting

$$y_R = B_R^{-T}(c_B - B_{R_c}^T y_{R_c}) \tag{2.20}$$

into the preceding leads to

$$N_R^T y_R + N_{R_c}^T y_{R_c} + z_N = c_N,$$

which implies that $\{y = (y_R^T, y_{R_c}^T)^T, z_N\}$ satisfies

$$N^T y + z_N = c_N.$$

In addition, (2.20) also implies that

$$B^T y = (B_R^T, B_{R_c}^T)(y_R^T, y_{R_c}^T)^T = c_B.$$

Therefore, the latter point is in D_N.

Conversely, assume that the latter point is in D_N. From the definition of D_N and (2.20), it follows that

$$\begin{aligned} z_N &= c_N - N^T y \\ &= c_N - N_R^T B_R^{-T}(c_B - B_{R_c}^T y_{R_c}) - N_{R_c}^T y_{R_c} \\ &= c_N - N_R^T B_R^{-T} c_B - (N_{R_c}^T - N_R^T B_R^{-T} B_{R_c}^T) y_{R_c}, \end{aligned}$$

which along with (2.18) leads to

$$\bar{N}_{R_c}^T y_{R_c} + z_N = \bar{c}_N.$$

Therefore, the former point is in D'_N, and the proof is complete. □

Lemma 2.1 establishes an 1-to-1 correspondence between points of D'_N and of D_N. In particular, it is obvious that $\{y_{R_c} = 0, z_N = \bar{c}_N \geq 0\} \in D'_N$ corresponds to

$$\{y_R = B_R^{-T} c_B, y_{R_c} = 0,\ z_B = 0, z_N = \bar{c}_N \geq 0\} \in D_N. \tag{2.21}$$

Thus, solving subproblem (2.17) amounts to maximizing the objective of (2.1) over D_N.

Theorem 2.1. *If B is a deficient-basis, then D_N is level dual face, as well as D'_N.*

Proof. Without loss of generality, assume that $k < m$.

Under the assumption, the residual Δy given by (2.13) is zero, and hence so is $\bar{b}_{R_c}$. Thus, D'_N is level dual face, over which the objective of (2.17) equals zero.

By Lemma 2.1, any $\{y_{R_c}, z_N \geq 0\} \in D'_N$ corresponds to

$$\{z_B = 0,\ y_R = B_R^{-T}(c_B - B_{R_c}^T y_{R_c}),\ y_{R_c},\ z_N \geq 0\} \in D_N.$$

Therefore, the objective value of the original dual problem (2.1) at the latter solution equals

$$(2.22)\qquad \begin{aligned} b_R^T y_R + b_{R_c}^T y_{R_c} &= b_R^T B_R^{-T}(c_B - B_{R_c}^T y_{R_c}) + b_{R_c}^T y_{R_c} \\ &= b_R^T B_R^{-T} c_B - y_{R_c}^T B_{R_c} B_R^{-1} b_R + y_{R_c}^T b_{R_c} \\ &= b_R^T B_R^{-T} c_B + y_{R_c}^T (b_{R_c} - B_{R_c} B_R^{-1} b_R). \end{aligned}$$

On the other hand, a comparison between the right-hands of (2.10) and (2.15) gives that

$$0 = \bar{b}_{R_c} = b_{R_c} - B_{R_c} B_R^{-1} b_R.$$

Combining the preceding and (2.22) leads to

$$b_R^T y_R + b_{R_c}^T y_{R_c} = b_R^T B_R^{-T} c_B,$$

which is constant over D_N, and the proof is complete. □

Corollary 2.1. *If $\bar{b}_{R_c}$ in tableau (2.10) vanishes, then the corresponding D_N is a level dual face.*

Nevertheless, it turned out that this approach coincides with one already used in [11, 12]. So, it appears to be blocked. Fortunately, the barrier can be broken by further partitions.

3. Dual search direction

In this section, we derive a fresh new dual search direction through a so-called *dual face subprogram.*

The trick is to partition further. Let N' be subset of N and R_1 subset of R_c, and set

$$(3.1)\qquad N = N \backslash N', \qquad R_c = R_c \backslash R_1, \qquad |N'| = |R_1| = k'.$$

In particular, assume $\bar{N}'_R = 0, \bar{N}'_{R_1} = I_{k'}$, and $\bar{N}'_{R_c} = 0$ so that tableau (2.10) can be written

(3.2)

x_B^T	$x_{N'}^T$	x_N^T	f	RHS
I_k		$\bar{N}_R$		$\bar{b}_R$
	$I_{k'}$	$\bar{N}_{R_1}$		$\bar{b}_{R_1}$
		$\bar{N}_{R_c}$		$\bar{b}_{R_c}$
	$\bar{c}_{N'}^{\mathrm{T}}$	$\bar{c}_N^{\mathrm{T}}$	-1	

which corresponds to dual face with $z_B = 0$ fixed.

Handled from the dual side, the preceding is termed *(canonical) dual face tableau*, although it still represents problem (1.1) from the primal side.

Using the active part of (3.2), construct the so-called *dual face subprogram* below:

$$\begin{array}{ll} \max & \bar{b}_{R_1}^T y_{R_1} + \bar{b}_{R_c}^T y_{R_c} \\ \text{s.t.} & \begin{pmatrix} I_{k'} & \\ \bar{N}_{R_1}^T & \bar{N}_{R_c}^T \end{pmatrix} \begin{pmatrix} y_{R_1} \\ y_{R_c} \end{pmatrix} + \begin{pmatrix} z_{N'} \\ z_N \end{pmatrix} = \begin{pmatrix} \bar{c}_{N'} \\ \bar{c}_N \end{pmatrix}, \qquad z_{N'}, z_N \geq 0. \end{array} \tag{3.3}$$

We will use the preceding to create a search direction to take one step forward.

Consider the following direction:

$$\Delta y_{R_1 \cup R_c} = \bar{b}_{R_1 \cup R_c}, \tag{3.4}$$

$$\Delta z_{N' \cup N} = \{-\bar{b}_{R_1}, -\bar{N}_{R_1}^T \bar{b}_{R_1} - \bar{N}_{R_c}^T \bar{b}_{R_c}\}. \tag{3.5}$$

Then, the following can be claimed.

Lemma 3.1. *If $\Delta y_{R_1\cup R_c}$ is nonzero, the direction defined by (3.4)-(3.5) is uphill in the null of the coefficient matrix of constraint equations, with respect to subprogram (3.3).*

Proof. It is uphill, since $\bar{b}_{R_1\cup R_c}$ is the gradient of the objective function. In addition, the following equation is easily verified:

$$\begin{pmatrix} I_{k'} & \\ \bar{N}_{R_1}^T & \bar{N}_{R_c}^T \end{pmatrix} \begin{pmatrix} \Delta y_{R_1} \\ \Delta y_{R_c} \end{pmatrix} + \begin{pmatrix} \Delta z_{N'} \\ \Delta z_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{3.6}$$

Therefore, the claim is valid. □

Lemma 3.1 ensures that the direction defined by (3.4)-(3.5) is eligible to be a search direction. In fact, $\{\bar{b}_{R_1}, \bar{b}_{R_c}\}$ is the steepest ascent direction in $\{y_{R_1}, y_{R_c}\}$-space.

In particular, we prefer matrix $B\cup N'$ being deficient-basis, because $\bar{b}_{R_c}$ vanishes in this case, simplifying formulas (3.4)-(3.5) as

$$\Delta y_{R_1\cup R_c} = \{\bar{b}_{R_1}, 0\}, \qquad \Delta z_{N'\cup N} = \{-\bar{b}_{R_1},\, -\bar{N}_{R_1}^T \bar{b}_{R_1}\}. \tag{3.7}$$

Furthermore, from a numerical point of view, it seems reasonable to use normalized $\bar{b}_{R_1}$ instead. This leads to the following search direction:

$$\Delta y_{R_1\cup R_c} = \{\bar{b}_{R_1}/\|\bar{b}_{R_1}\|_2, 0\}, \qquad \Delta z_{N'\cup N} = \{-\Delta y_{R_1}, -\bar{N}_{R_1}^T \Delta y_{R_1}\}. \tag{3.8}$$

4. Update of dual feasible solution

Assume that search direction $\Delta z_{N'\cup N}$ was determined by (3.5).

Introduce index set

$$J = \{j \in N'\cup N \mid \Delta z_j < 0\}. \tag{4.1}$$

If J is nonempty, the current dual feasible solution to (3.3), i.e.,

$$y_{R_1\cup R_c} = 0, \qquad z_{N'\cup N} = \bar{c}_{N'\cup N} \geq 0 \tag{4.2}$$

can be updated by the following formula:

$$\hat{y}_{R_1 \cup R_c} = \beta \bar{b}_{R_1 \cup R_c}, \tag{4.3}$$

$$\hat{c}_{N' \cup N} = \bar{c}_{N' \cup N} + \beta \Delta z_{N' \cup N}, \tag{4.4}$$

where β is the maximum possible step-size that can be achieved, i.e.,

$$\beta = \min_{j \in J} \; -\bar{c}_j / \Delta z_j \geq 0. \tag{4.5}$$

Note. Applying (3.7) and (4.4) together amounts to adding $-\beta \bar{b}_{R_1}$ times of rows corresponding to R_1 to the bottom row in the dual face tableau.

Dual feasible solution is said *degenerate* if some component of $\bar{c}_J$ is 0. In this case, dual step-size β vanishes, and the "new" solution is not really new, but the same as the old.

Using the preceding notation, we claim the following result.

Theorem 4.1. *If J defined by (4.1) is nonempty, then (4.3)-(4.4) along with (4.5) give a new dual feasible solution to subprogram (3.3).*

Proof. Note that β is well-defined by (4.5) under the nonempty assumption on J.

Since old solution (4.2) satisfies the equality constraint of (3.3) and direction (3.4)-(3.5) satisfies (3.6), the new solution satisfies the equality constraint.

By (4.5), it holds that

$$\beta \leq -\bar{c}_j / \Delta z_j, \qquad j \in J. \tag{4.6}$$

Besides, (4.1) ensures that

$$\Delta z_j < 0, \qquad j \in J.$$

Thus, multiplying (4.6) by Δz_j gives

$$\beta\Delta z_j \geq -\bar{c}_j, \qquad j \in J.$$

which along with (4.4) leads to

$$\hat{c}_j = \bar{c}_j + \beta\Delta z_j \geq 0, \qquad j \in J. \tag{4.7}$$

In case of $(N' \cup N)\backslash J \neq \emptyset$. on the other hand, (4.1) implies that

$$\Delta z_j \geq 0, \quad j \in (N' \cup N)\backslash J,$$

combining which, $\bar{c}_{N'\cup N} \geq 0$ and $\beta \geq 0$ results in

$$\hat{c}_j = \bar{c}_j + \beta\Delta z_j \geq 0, \qquad j \in (N' \cup N)\backslash J. \tag{4.8}$$

Therefore, it holds that $\hat{c}_{N'\cup N} \geq 0$, and the proof is complete. □

Theorem 4.2. *If search direction is determined by (3.4)-(3.5), the new solution (4.3)-(4.4) increases the dual objective value by*

$$\Delta f = \beta(\|\bar{b}_{R_1}\|_2^2 + \|\bar{b}_{R_c}\|_2^2). \tag{4.9}$$

with respect to dual subprogram (3.3).

Proof. By (4.3), the old zero-valued solution in $y_{R_1\cup R_c}$ space is updated to the new solution $\beta\bar{b}_{R_1\cup R_c}$, and hence the corresponding objective value increases from zero up to $\beta\bar{b}^T_{R_1\cup R_c}\bar{b}_{R_1\cup R_c}$, which is equal to (4.9). □

It is noticeable that if search direction (3.7) is used instead, the corresponding objective increment becomes $\beta\|\bar{b}_{R_1}\|_2^2$, whereas the increment becomes $\beta\|\bar{b}_{R_1}\|_2$ if direction (3.8) used.

Note. In practice, it does not make sense to update $\bar{y}$. If needed, the final dual solution in y-space can be calculated cheaply at the end of solution process, as will be clear in Section 6.

As for J being empty, a case when step-size β is not well-defined, we are still done as the following says.

Theorem 4.3. *Assume that $\bar{b}_{R_1 \cup R_c}$ is nonzero. If $J = \emptyset$, problem (1.1) is infeasible.*

Proof. From the last half of the proof of Theorem 4.1, in case of $J = \emptyset$, new solution (4.3)-(4.4) are dual feasible to (3.3) for all $\beta > 0$, . Under condition $\bar{b}_{R_1 \cup R_c} \neq 0$, the corresponding objective increment, given by Theorem 4.2, goes to ∞, as so is β.

Therefore, dual subprogram (3.3) is upper unbounded, and hence there exists no feasible solution to the corresponding primal subprogram, as well as to problem (1.1). □

5. Pivoting operation

In this section, we discuss pivoting operations and associated index set adjustment.

Assume that new dual solution $\hat{c}_{N' \cup N}$ was determined by (4.4) along with (4.5). Introduce index set

$$J' = \arg\min_{j \in J} \; -\bar{c}_j / \Delta z_j. \tag{5.1}$$

where J is defined by (4.1). It is clear that for each $j \in J'$, component $\hat{c}_j$ reaches the critical point zero. Related variables or indices are said *blocking* ones.

There are two types of iterations as follows:

(A) $\tilde{J} \stackrel{\triangle}{=} N' \cap J'$ is nonempty (or $J' \not\subset N$).

There is then no need for pivoting, and the involved computational cost is quite cheap. Before going to the next iteration, what to do is only to execute the following tactic .

Tactic 5.1. *Assume that row index set* $\tilde{I}$ *corresponds to* $\tilde{J}$*. Update* $\{N', B,\ R_1, R, k\}$ *by*

$$N' = N' \backslash \tilde{J}, \qquad B = B \cup \tilde{J}, \tag{5.2}$$
$$R_1 = R_1 \backslash \tilde{I}, \qquad R = R \cup \tilde{I}, \qquad k = k + |\tilde{J}|.$$

(B) $\tilde{J} \stackrel{\triangle}{=} N' \cap J'$ is empty (or $J' \subset N$).

There is a need for pivoting. To do so, execute the following rule first.

Rule 5.1. *[Column rule] Select column index* q *such that*

$$q \in \arg\min_{j \in J'} \Delta z_j. \tag{5.3}$$

Then determine row index p such that

$$p \in \arg\max_{i \in R_c} |\bar{a}_{i\,q}|. \tag{5.4}$$

There will be following two cases arising:

(i) $\bar{a}_{p\,q} \neq 0$.

We update the tableau by using $\bar{a}_{p\,q}$ as pivot (see below), and then update index sets conformably by the following tactic.

Tactic 5.2. *Update* $\{N, B,\ R_c, R, k\}$ *by*

$$N = N \backslash q, \qquad B = B \cup q, \tag{5.5}$$
$$R_c = R_c \backslash p, \qquad R = R \cup p, \qquad k = k + 1.$$

(ii) $\bar{a}_{p\,q} = 0$.

There are multiple choices for row pivoting. We bring up the following plausible one.

Rule 5.2. *[Row rule] Assume that column index q was determined by (5.3). Reselect row index p such that*

$$p \in \arg\min\{-\bar{b}_i \bar{a}_{i,\,q} \mid i \in R_1\}, \tag{5.6}$$

corresponding to $j_p \in N'$.

Using the preceding notation, we have the following result.

Proposition 5.1. *Δz_q and $-\bar{b}_p \bar{a}_{p,\,q}$ have the same minus sign.*

Proof. It is evident that $\bar{a}_{i'\,q} = 0$ implies

$$\bar{N}_{R_c,\,q} = 0. \tag{5.7}$$

In addition, by (4.1) and $q \in J$, it is known that $\Delta z_q < 0$. Hence, combining (3.5) and (5.7) leads to

$$\Delta z_q = -\bar{N}^T_{R_1,\,q} \bar{b}_{R_1} - \bar{N}^T_{R_c,\,q} \bar{b}_{R_c} = \sum_{i \in R_1} -\bar{b}_i \bar{a}_{i,q} < 0.$$

The preceding along with (5.6) gives $-\bar{b}_p \bar{a}_{p,\,q} < 0$, which has the same minus sign as Δz_q. □

The preceding ensures pivot $\bar{a}_{p,q}$ with a nonzero value. It is more than that. It also implies that Rule 5.2 conforms to the heuristic characteristic of the optimal solution, in dual context (section 2.5, [15]).

Once a pivot determined, what to do next is to update the tableau. To do so, modify the first m rows by premultiplying the p-th row by $1/\bar{a}_{p,q}$, and adding $-\bar{a}_{i,q}$ times of the row to the i-th row, for $i \in R_1 \cup R_c,\ i \neq p$.

Then, update index sets by the following tactic conformably.

Tactic 5.3. *Update* $\{N, B, N', R_1,\ R, k\}$ *by*

$$(5.8)\qquad \begin{array}{llll} N = N\backslash q, & B = B \cup q, & N' = N'\backslash\{j_p\}, & \\ R_1 = R_1\backslash p, & R = R \cup p, & N = N \cup \{j_p\}, & k = k+1. \end{array}$$

We ignore related column and row permutations in conjunction with the tableau. After all, it is not necessary to move around entire column or row of entries in the implementation.

6. Optimality condition

Let (3.2) be current dual face tableau. It is noted that if $k = m$, none of formulas (3.5), (3.7) and (3.8) is well-defined, and if $k < m$ but $\bar{b}_{R_1 \cup R_c} = 0$, all these directions vanish.

Consider the following solution given by tableau (3.2):

$$(6.1)\qquad \bar{x}_B = \bar{b}_R, \qquad \bar{x}_{N' \cup N} = 0.$$

Clearly, if B is not a deficient-basis, such a solution does not satisfy the equality constraint, let alone being a feasible solution to (1.1).

Lemma 6.1. *If B is a deficient-basis, solution (6.1) satisfies the equality constraint of (1.1), corresponding to primal objective value $\bar{f}$.*

Proof. Note that all resulting tableaus represent the same problem (1.1). It is obvious that B is deficient-basis if and only if $k = m$ or $k < m$ but $\bar{b}_{R_1 \cup R_c} = 0$. Under the assumption, it is easily verified that solution (6.1) satisfies the equality constraint of

(3.2), as well as that of (1.1), with the right-most entry of the bottom row giving the related objective value $\bar{f}$. □

Denote by $D_{N'\cup N}$ the dual face defined by (2.2) with N replaced by $N' \cup N$.

Lemma 6.2. *If B is a deficient-basis, $D_{N'\cup N}$ is a level dual face.*

Proof. Validity of the statement is obvious through Corollary 2.1. □

Theorem 6.1. *Assume that B is a deficient-basis. If the following index set*

$$T = \{j_i \in B \mid \bar{b}_i < 0,\ i \in R\} \tag{6.2}$$

is empty, then $D_{N'\cup N}$ is a dual optimal face with a pair of primal and dual optimal solutions.

Proof. Lemma 6.1 and emptiness of T together imply that $\bar{x}$ given by (6.1) is a primal feasible solution to (1.1). On the other hand, it is obvious that $\{y_{R_1\cup R_c} = 0, z_{N'\cup N} = \bar{c}_{N'\cup N} \geq 0\}$ corresponds to

$$\{y_R = B_R^{-T} c_B, y_{R_1\cup R_c} = 0,\ z_B = 0, z_{N'\cup N} = \bar{c}_{N'\cup N} \geq 0\} \in D_{N'\cup N}, \tag{6.3}$$

which is a dual feasible solution. The preceding and $\bar{x}$ exhibit complementarity slackness, and are therefore a pair of primal and dual optimal solutions. According to Lemma 6.2, $D_{N'\cup N}$ is a dual level face. Including the dual optimal solution, it is a dual optimal face. □

In the case of $T \neq \emptyset$ when optimality cannot be declared, execute the following tactic, and go on the next iteration.

Tactic 6.1. *Using notation* $T' = \{i \in R \mid \bar{b}_i < 0\}$*, update* $\{B, N', R, R_1, k\}$ *by*

$$\begin{aligned} &B = B\backslash T, \qquad N' = N' \cup T, \\ &R = R\backslash T', \qquad R_1 = R_1 \cup T', \qquad k = k - |T|. \end{aligned} \tag{6.4}$$

Consequently, we are faced with an expanded new dual face. Note that the zero-valued new $\bar{c}_{N'}$ will never affect the determination of the next step-size, since the new $\bar{b}_{R_1}$ is nonpositive, and hence $\Delta z_{N'} = -\bar{b}_{R_1}$ is nonnegative.

7. Dual face algorithm: Tableau form

Based on discussions made in previous sections, we formulate the dual face algorithm in tableau form in this section.

In each iteration, the dual face tableau is updated with $\bar{f}$ increasing. A series of iterations are performed until $k = m$ or $\bar{b}_{R_1 \cup R_c} = 0$ when a dual level face is reached (unless upper unboundedness detected). At the end of such a series, referred to as *dual face contraction*, optimality is tested: if set T is empty, optimality is achieved. In the other case, the dual face is expanded, and the next dual face contraction is carried out.

Although solution process can be initiated with any pseudo-basis, we prefer a deficient-basis since this will make $\bar{b}_{R_c}$ vanish throughout the whole solution process. Therefore, the simpler search direction formula (3.7) will be used.

Let B be initial deficient-basis. We take the following partition to get started:

$$\begin{aligned} &N = A\backslash B \qquad N' = \emptyset, \\ &R = R, \qquad R_1 = \emptyset, \quad k \leq m. \end{aligned} \tag{7.1}$$

Overall steps are summarized into the following model.

Algorithm 1.[Dual face algorithm: tableau form] Initial : dual face tableau in form (3.2) with partition (7.1) and $\bar{b}_{R_c} = 0$. This algorithm solves problem (1.1).

1. If $k = m$ or $\bar{b}_{R_1} = 0$, then:

   (1) stop if T defined by (6.2) is empty;

   (2) execute Tactic 6.1.

2. Compute $\Delta z_{N' \cup N}$ by (3.7).

3. Stop if J defined by (4.1) is empty.

4. Determine β by (4.5).

5. If $\beta > 0$, update $\bar{c}_{N' \cup N}$ by (4.4).

6. If $\tilde{J} = N' \cap J' \neq \emptyset$, where J' is defined by (5.1),

   execute Tactic 5.1, and go to step 1.

7. Determine column index q by (5.3) and row index p by (5.4).

8. If $\bar{a}_{p,\,q} = 0$, redetermine row index p by (5.6).

9. Prewmultiply the p-th row by $1/\bar{a}_{p,q}$, and then add $-\bar{a}_{i,q}$ times of the row to

   the i-th row, for $i \in R_1 \cup R_c,\ i \neq p$.

10 If $p \in R_c$, execute Tactic 5.2; else execute Tactic 5.3.

11. Go to step 1.

Theorem 7.1. *If each dual face contraction involves a non-degenerate dual feasible solution, Algorithm 1 terminates either at*

(i) step 1(1), giving a pair of primal and dual optimal solutions; or at

(2) step 3, declaring primal infeasibility.

Proof. It is seen that k increases by 1, at least, in each iteration. Therefore, within finitely many iterations, either k increases to m or $\bar{b}_{R_1 \cup R_c}$ vanishes so that each dual face contraction ends with a level dual face (Lemma 6.2), unless detecting primal infeasibility at step 3, by Theorem 4.3.

Assume that the algorithm does not terminate. Then, failing for optimality test, it performs infinitely many dual face contractions, each includes a nonzero step-size. Consequently, objective value $\bar{f}$, corresponding to the end level dual face, increases strictly (Theorem 4.2), as contradicts that the number of dual level faces is finite. Therefore, the algorithm terminates, with outlet step 1(1) giving a dual optimal face together with a pair of dual and primal optimal solutions, by Theorem 6.1. □

8. Dual face algorithm: Revised form

In this section, we convert Algorithm 1 into a revised form using LU factorization.

Introduce symbol $\widehat{B} \triangleq B \cup N'$. Let current $\bar{c}_{N' \cup N}$, $\tilde{b}$, and LU factorization below be available:

$$\widehat{B}_{R \cup R_1} = LU. \tag{8.1}$$

The following quantity expressions required can be derived based on a comparison between (2.10) and (2.15), with B and R replaced respectively by $B \cup N'$ and $R \cup R_1$, e.g., giving

$$\bar{N}_{R \cup R_1} = \widehat{B}^{-1}_{R \cup R_1} N_{R \cup R_1}, \qquad \bar{N}_{R_c} = N_{R_c} - \widehat{B}_{R_c} \widehat{B}^{-1}_{R \cup R_1} N_{R \cup R_1}, \tag{8.2}$$

which are useful for obtaining the search direction.

(A) Search direction $\Delta z_{N' \cup N} = \{-\bar{b}_{R_1}, -\bar{N}^T_{R_1} \bar{b}_{R_1} - \bar{N}^T_{R_c} \bar{b}_{R_c}\}$ (3.5).

(i) $t_1^T \triangleq \bar{b}^T_{R_1} \bar{N}_{R_1} = (0^T_R, \bar{b}^T_{R_1}) \widehat{B}^{-1}_{R \cup R_1} N_{R \cup R_1}$.

Solve following triangular systems for $v = \widehat{B}_{R\cup R_1}^{-T}(0_R^T, \bar{b}_{R_1}^T)^T$:

$$U^T u = (0_R^T, \bar{b}_{R_1}^T)^T, \qquad L^T v = u, \tag{8.3}$$

and then compute

$$t_1 = N_{R\cup R_1}^T v. \tag{8.4}$$

(ii) $t_2 \triangleq \bar{N}_{R_c}^T \bar{b}_{R_c} = (N_{R_c} - \widehat{B}_{R_c}\widehat{B}_{R\cup R_1}^{-1} N_{R\cup R_1})^T \bar{b}_{R_c}$ (ignore if $\bar{b}_{R_c} = 0$).

Solve the following systems for $w = \widehat{B}_{R\cup R_1}^{-T}\widehat{B}_{R_c}^T \bar{b}_{R_c}$ first:

$$U^T u = \widehat{B}_{R_c}^T \bar{b}_{R_c}, \qquad L^T w = u, \tag{8.5}$$

and then compute

$$t_2 = N_{R_c}^T \bar{b}_{R_c} - N_{R\cup R_1}^T w. \tag{8.6}$$

Finally, it follows that

$$\Delta z_N = \{-\bar{b}_{R_1}, -t_1 - t_2\}. \tag{8.7}$$

(B) pivot sub-columns $\{\bar{a}_{R\cup R_1,q},\ \bar{a}_{R_1,q},\ \bar{a}_{R_c,q}\}$.

Solve the following triangular systems for $w = \bar{a}_{R\cup R_1,q} = \widehat{B}_{R\cup R_1}^{-1} a_{R\cup R_1,q}$:

$$Lu = a_{R\cup R_1,q}, \qquad Uw = u. \tag{8.8}$$

Then it follows that

$$\bar{a}_{R_1,q} = w_{R_1}, \qquad \bar{a}_{R_c,q} = a_{R_c,q} - \widehat{B}_{R_c} w. \tag{8.9}$$

(C) Update of right-hand side $\bar{b}$ (ignore if $p \in R_c$ and $\bar{b}_{R_c} = 0$).

Determine $\alpha = \bar{b}_p/\bar{a}_{p,q}$, and update

$$\hat{b} = \bar{b} - \alpha\bar{a}_q, \qquad \hat{b}_p = \alpha. \tag{8.10}$$

As for updating LU factors of $\widehat{B}_{R\cup R_1}$, nothing needs to do with Tactic 5.1 or 6.1, but the following two cases arising:

Case 1. *Rank increasing with Tactic 5.2 :* rank k *of* $\widehat{B}_{R\cup R_1}$ *increases by* 1. *Refer to Section 20.5.2 in* [15].

Case 2. *Rank remaining with Tactic 5.3: k remains unchanged. The related updating is the same as in conventional simplex context, in essence.*

Now Algorithm 1 can be converted into the following model.

Algorithm 2. [Dual face algorithm: revised form] Initial : deficient-basis B, partition (7.1), $\widehat{B}_{R\cup R_1} = LU$, $\bar{b}_{R\cup R_1}$ and $\bar{c}_{N'\cup N}$. This algorithm solves problem (1.1).

1. Go to step 4 if $k = m$.
2. Go to step 6 if $\Delta z_{N'} = -\bar{b}_{R_1} \neq 0$.
3. Stop if T defined by (6.2) is empty (optimality achieved).
4. Execute Tactic 6.1, and go to step 2.
5. Solve systems (8.3) for v, and compute $\Delta z_N = -N^T_{R\cup R_1}v$.
6. Stop if J defined by (4.1) is empty (infeasible problem).
7. Determine β by (4.5).
8. If $\beta > 0$, update $\bar{c}_{N'\cup N}$ by (4.4).
9. If $\tilde{J} = N' \cap J' \neq \emptyset$, where J' is defined by (5.1), execute Tactic 5.1,

   and go to step 1.

10. Determine column index q by (5.3).

11. Solve systems (8.8) for $w = \bar{a}_{R\cup R_1,q}$.

12. Compute $\bar{a}_{R_c,q}$ by the second expression of (8.9), and determine p by (5.4).

13. If $\bar{a}_{p,q} \neq 0$, then

    (1) update L, U by Case 1, and execute Tactic 5.2;

else

    (2) redetermine p (with j_p) by (5.6);

    (3) update L, U by Case 2;

    (4) update $\bar{b}_{R\cup R_1}$ by (8.10), where $\alpha = \bar{b}_p/\bar{a}_{p,q}$, and execute Tactic 5.3.

14. Go to step 1.

9. Final remarks

If the proposed algorithm gets started from a pseudo-basis rather than deficient-basis, one should use search direction (3.5) instead of (3.7), since nonzero $\bar{b}_{R_c}$ then contributes extra growth $\beta\|\bar{b}_{R_c}\|_2^2$ to the dual objective increment, by Theorem 4.2. Accordingly, however, two more triangular systems have to be solved. It might be a good idea to get started with a pseudo-basis of small k, proceed with $N', R_1 = \emptyset$ and search direction

$$\Delta z_N = -\bar{N}_{R_c}^T(\bar{b}_{R_c}/\|\bar{b}_{R_c}\|_2), \tag{9.1}$$

until $\bar{b}_{R_c}$ vanishes, and go over to Algorithm 2 with direction (3.7) instead.

Alternatively, instead of the whole $\bar{b}_{R_1}$, it is possible to use a part of it to form search direction, e.g., such that

$$|\bar{b}_i| \geq \rho, \qquad i \in R_1, \tag{9.2}$$

where ρ is a threshold, decreasing in solution process.

Surprisingly enough, it turns out that the new algorithm and the primal face algorithm [17] just proposed are a pigeon pair. It is interesting to compare the two algorithms. It seems to be natural to use the latter to achieve dual feasibility, and then use the former to achieve optimality, or vice versa (sections 13.4 and 14.4, [15]). It is noted that both algorithms can be implemented, the same as in the conventional simplex context.

An advantage of the new dual face algorithm is that it solves small systems, compared with those solved by simplex algorithms, while it handles two systems as well. In a single iteration, therefore, overall computational cost by the former is less than the latter.

Another advantage of the new algorithm is its good stability, compared with simplex algorithms. The former tends to select a large pivot in magnitude, while the latter's pivot can be arbitrarily close to zero. As we all know, the latter sometimes fails to maintain a usable basis, and has to restart from scratch.

Finally, the most attractive feature of the proposed algorithm might be the dual objective increment $\beta\|\bar{b}_{R_1}\|_2^2$ achieved per iteration. It is expected that this would significantly reduce the number of iterations required. At this stage, however, there are no numerical results available though. We even do not know which search direction is better in practice. The implementation of the new algorithm and computational experiments are expected.

References

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

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

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

[4] G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, 1963.

[5] G.H. Golub, Numerical methods for solving linear least squares problems,*Numer.Math.*, **7** (1965), 206-216.

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

[7] P.-Q. Pan, A dual projective simplex method for linear programming, *Computers and Mathematics with Applications*, **35** (1998a), 119–135.

[8] P.-Q. Pan, A basis-deficiency-allowing variation of the simplex method, *Computers and Mathematics with Applications* , **36** (1998), 33-53.

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

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

[11] P.-Q. Pan, A dual projective pivot algorithm for linear programming, *Computational Optimization and Applications*, **29** (2004), 333-344.

[12] P.-Q. Pan, A revised dual projective pivot algorithm for linear programming, *SIAM Journal on Optimization*, **16** (2005), 49-68.

[13] P.-Q. Pan, A primal deficient-basis algorithm for linear programming, *Applied Mathematics and Computation*, **198** (2008b), 898-912.

[14] P.-Q. Pan, An affine-scaling pivot algorithm for linear programming, *Optimization*, **62** (2013), 431-445.

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

[16] P.-Q. Pan, Variant of the dual face algorithm using Gauss-Jordan elimination for linear programming, Journal of the Operations Research Society of China, **4** (2016), 347-356.

[17] P.-Q. Pan, A new face algorithm using LU factorization for linera programming, preprint, 2020.

[18] M. A. Saunders, *Large Scale Linear Programming Using the Cholesky Factorization*, Stanford University Report STAN-CS-72-152, 1972.

[19] Y.-y Shi, L.-H. Zhang and W.-X. Zhu, A review of Linear Programming Computation by Ping-Qi Pan, *European Journal of Operational Research*, **267** (2018) No. 3, 1182-1183.

Department of Mathematics, Southeast University, Nanjing 210096, China

E-mail address: panpq@seu.edu.cn