RUTCOR
RESEARCH
REPORT

On a Dual Method for a Specially Structured Linear Programming Problem

Olga Fiedler [a] András Prékopa [b]
Csaba I. Fábián [c]

RRR 25-95, June 1995

RUTCOR • Rutgers Center for Operations Research • Rutgers University • P.O. Box 5062 • New Brunswick New Jersey • 08903-5062
Telephone: 908-445-3804
Telefax: 908-445-5472
Email: rrr@rutcor.rutgers.edu

[a] ofiedler@math.fu-berlin.de, Freie Universität Berlin, FB Mathematik u. Informatik, Arnimalle 2-6, 14195 Berlin, Germany

[b] prekopa@rutcor.rutgers.edu, RUTCOR, Rutgers Center for Operations Research, New Brunswick, New Jersey 08903-5062, USA

[c] fabian@cs.elte.hu, Visitor to RUTCOR and Dept. of OR, L.Eötvös University, Muzeum krt. 6-8, Budapest, Hungary, 1088

Rutcor Research Report
RRR 25-95, June 1995

On a Dual Method for a Specially Structured Linear Programming Problem

Olga Fiedler   András Prékopa   Csaba I. Fábián

Abstract. The paper revises and improves on a Dual Type Method (DTM) developed by A. Prékopa (1990) in two ways. The first improvement allows us, in each iteration, to perform the largest step toward the optimum. The second inprovement consists of exploting the *structure* of the *working basis*, which has to be inverted in each iteration of the DTM, and updating its inverse in product form, as it is usual in case of the standard dual method.

Acknowledgements: The first author gratefully acknowledge the support of the *Kommission für die Förderung von Nachwuchswissentschaftlerinnen der FU Berlin.*

The third author gratefully acknowledges the support of the Foundation for the Development of Higher Education in Hungary, project III/1141.

0. Introduction

1990 A. Prékopa developed the Dual Type Method (DTM) to solve a specially structured linear programming problem (see [6]). This method can be applied to solve Stochastic Programming Problems, where the underlying problem is an LP and some of the right hand side values are discrete random variables; the violations of the random constraints are penalized by piecewise linear and convex functions added to the original objective function. It can also be applied to solve special linear programming problems, e.g., the constrained minimum absolute deviation problem (see [8]).

In this paper we present an *Improved* Dual Type Method (IDTM) for the solution of the same kind of programming problems as in [6]. The novelty here is that, in each iteration, the *largest* possible increase is made toward the optimum value and the inverse of the working basis is updated in the product form.

In each iteration of the *standard* Dual Method (DM) (see [3]) the inverse of the new basis (obtained from the old one by a column-exchange) is updated by the use of simple transformation formulas. In the DTM the inversion of the dual feasible basis $\hat{B}$ reduces (due to its *block*-structure which arises from the use of the λ-representation) to the problem of inversion of a smaller size matrix, called *working basis*, B which is taken from the intersection of the 0th block and $\hat{B}_1$ (cf. (1.6)). The transformation formulas to find the inverse of the *new* working basis B_1 are, however, more complicated than those in the standard DM, because B_1 arises from the old working basis B not just by a simple column-exchange, but by the application of more complicated rules. This yields the invertion of the new working basis B_1 in each iteration of the DTM. In the IDTM we exploit the structure of the new working basis in such a way that we are able to update its inverse (similarly as in the DM) in product form (cf., e.g., (2.1.11), (2.2.9), (2.4.7)). We also give simple transformation formulas to compute the basic solution and the dual vector corresponding to the new dual feasible basis $\hat{B}_1$.
The Algorithm for the IDTM is summarized in Section 3. The implementation of the IDTM, based on C.I. Fábián's general optimization routine library called LINX, is described in Section 4. The code is underway for full implementation.

1. Formulation of the Problem

The IDTM solves the problem

$$
\text{(1.1)}\qquad \begin{array}{ll} \min_x & \{c^T x + \sum_{i=1}^r f_i(y_i)\} \\ \text{s.t.} & Ax = b \\ & T_i x = y_i,\ i = 1,\dots,r \\ & x \ge 0 \end{array}
$$

where $A = (a_1, \dots, a_n)$ is an $(m \times n)$, T is an $(r \times n)$ matrix, $T_1, \dots, T_r$ are the rows of T, and the vectors x, b, y are of suitable sizes. The $f_i(\cdot)$ are piecewise linear and convex functions defined in the intervals $[z_{i0}, z_{ik_i+1}]$ with breakpoints at $z_{i0} < z_{i1} < \dots < z_{ik_i+1}$, $i = 1, \dots, r$. For each $i = 1, \dots, r$ we will only need the function values $f_i(z_{ij}),\ j = 0, \dots, k_i + 1$. The convexity of $f_i(\cdot)$ holds if and only if the second order divided differences of the sequence $f_i(z_{ij}),\ j = 0, \dots, k_i + 1,\ i = 1, \dots, r$ are positive.

Given a discrete function f with values $f_j = f(z_j), j = 0, \dots, k+1$, its first and second order divided differences are defined by the formulas

$$
\text{(1.2)}\qquad \begin{array}{l} [z_j, z_{j+1}; f] := \frac{f_{j+1} - f_j}{z_{j+1} - z_j} \ ,\ j = 0, \dots, k, \\ [z_j, z_{j+1}, z_{j+2}; f] := \frac{[z_{j+1}, z_{j+2}\ ;\ f] - [z_j, z_{j+1}\ ;\ f]}{z_{j+2} - z_j} \ , \quad j = 0, \dots, k-1. \end{array}
$$

Using the λ-representation for the functions $f_i(\cdot)$, problem (1.1) can be reformulated as follows:

$$
\text{(1.3)}\qquad \begin{array}{ll} \min_{x,\lambda} & \{c^T x + \sum_{i=1}^r \sum_{j=0}^{k_i+1} f_i(z_{ij})\lambda_{ij}\} \\ \text{s.t.} & Ax = b \\ & T_i x = y_i \\ & \sum_{j=0}^{k_i+1} z_{ij}\lambda_{ij} = y_i \\ & \sum_{j=0}^{k_i+1} \lambda_{ij} = 1 \\ & x \ge 0,\ \lambda_{ij} \ge 0,\ i = 1, \dots, r\ , \quad j = 0, \dots, k_i + 1\ . \end{array}
$$

We assume that $z_{i0} \le T_i x \le z_{ik_i+1}$ $(i = 1, \dots, r)$ holds for any x that satisfies $Ax = b,\ x \ge 0$. If we introduce the notation $f_i(z_{ij}) = c_{ij}$, then (1.3) can be written in the following form:

$$
\text{(1.4)}\qquad \begin{array}{ll} \min_{x,\lambda} & \{c^T x + \sum_{i=1}^r \sum_{j=0}^{k_i+1} c_{ij}\lambda_{ij}\} \\ s.t. & Ax = b \\ & T_i x - \sum_{j=0}^{k_i+1} z_{ij}\lambda_{ij} = 0 \\ & \sum_{j=0}^{k_i+1} \lambda_{ij} = 1 \\ & x \ge 0,\ \lambda_{ij} \ge 0,\ j = 0, \dots,\ k_i + 1,\ i = 1, \dots r. \end{array}
$$

The matrix of the equality constraints is subdivided into $(r + 1)$ blocks:

$$
c_1 \dots c_n \quad c_{10} \dots c_{1k_1+1} \quad \dots \quad c_{r_0} \dots c_{rk_r+1}
$$

$$
(1.5)\qquad
\begin{pmatrix}
A & & & & \\
T_1 & -z_{10}\ldots-z_{1k_1+1} & & & \\
\vdots & & \ddots & & \\
T_r & & & -z_{r_0}\ldots-z_{rk_r+1} & \\
0 & 1\ldots1 & & & \\
\vdots & & \ddots & & \\
0 & & & 1\ldots\quad 1 &
\end{pmatrix}
$$

$$
\begin{array}{cccc}
0^{th}\text{ block} & 1^{st}\text{ block} & \ldots & r^{th}\text{ block}
\end{array}
$$

In [6] it is shown that an initial dual feasible basis for the problem (1.4) consists of those vectors, which trace out a (so-called) *working basis* $B = (a_{i1}, \ldots, a_{im})$ from A, the (corresponding to B) matrix T_B from T in the 0th block, and the consecutive pairs of vectors in the other blocks; the subscripts of the vectors in B are chosen in a special way (which is explained below), whereas the subscripts of the consecutive vectors in the blocks $1, \ldots, r$ are chosen arbitrarily. Thus

$$
(1.6)\qquad
\hat{B} = \begin{pmatrix}
B & & & & & \\
T_{1B} & -z_{1j_1} & -z_{1j_1+1} & & & \\
\vdots & & & \ddots & & \\
T_{rB} & & & & -z_{rj_r} & -z_{rj_r+1} \\
0 & 1 & 1 & & & \\
\vdots & & & \ddots & & \\
0 & & & & 1 & 1
\end{pmatrix},
$$

where T_{iB} , denotes the ith-row of T_B , $i = 1, \ldots, r$. The concept of "working basis" was first introduced by R. Wets in [9]. During the later iterations the basis structure of $\hat{B}$ may change in such a way that the working basis B may have s $(0 \leq s \leq r)$ more columns intersecting the matrix A, and s more rows from the matrix T. In this case there are exactly $(r - s)$ blocks with a consecutive pair in each block, and s blocks with a single vector in each block.

Let S be the set of those row subscripts of T corresponding to which only one $-z_{ij}$ is in a basic column, and $Q := \{1, \ldots, r\} \backslash S$. Then, the working basis takes the following form:

$$
(1.7)\qquad B = \begin{pmatrix} A_B \\ T_{iB}\ (i \in S) \end{pmatrix}, \quad \text{where} \quad A_B = (a_{i_1}, \ldots, a_{i_m}, a_{i_{m+1}}, \ldots, a_{i_{m+s}}) .
$$

The dual vector corresponding to any basis $\hat{B}$ for the problem (1.4) will be partitioned as (y_B^T, v^T, w^T), where y_B^T is an m- component vector and v, w are r-component vectors.

In the DTM the problem to *update the inverse* of the new dual feasible basis $\hat{B}_1$ in each step reduces (due to its special block-structure) to the problem of inverting *only* a working basis

B_1, which is of much smaller size than the initial dual feasible basis $\hat{B}$. Unfortunately, the transformation formulas for updating the inverse of the new working basis B_1 (which are usually used in simplex type methods) cannot always be applied here in a straightforward manner, because B_1 sometimes arises from the old working basis B not just by a simple column exchange, but by the use of more complicated rules, such as:

(1) extension of B by one additional column and one additional row; this happens when a column from one of the last r blocks (with two consecutive vectors in the basis) leaves the basis, and a nonbasic column from the 0th-block enters the basis, (cf. Case(i), Section 2);

(2) exchange of one row in B accompanied by a proper change in the coefficients of the cost function; this happens when a column from a block with two vectors in it leaves the basis, and a column from a block with just one vector in it enters the basis, (cf. Case(ii), Section 2);

(3) deletion of one column and one row from B; this happens when a column from the 0th-block leaves the basis and the entering column is from one of the last r blocks with just one vector (containing $-z_{ij_i}$) in a basis, (cf. Case(iv), Section 2).

Therefore, in the DTM, one has to invert the new working basis B_1 in each step. In the IDTM we are able to construct in each step an auxiliary matrix (with known inverse) such that the new working basis will be obtained from this auxiliary matrix by a simple column-exchange. This allows us for *updating in product form.*

Before we start with the presentation of the IDTM, which is a straightforward extension of the DTM, for the reader's convenience we give now a short description of an iteration of the DTM (cf. [6], pp. 448-454):

Step 0. Construction of an initial dual feasible basis $\hat{B}$:
i.) For $i = 1, \dots, r$ choose j_i satisfying $0 \le j_i \le k_i$. Include the columns with the subscripts (i, j_i), $(i, j_i + 1)$, $i = 1, \dots, r$, from the last r blocks of (1.5) into the basis $\hat{B}$.

ii.) Determine v_i and w_i ($i = 1, \dots, r$) by the equations

$$\left.\begin{aligned} -z_{ij_i} v_i + w_i &= c_{ij_i} \\ -z_{ij_i+1} v_i + w_i &= c_{i_{j_i+1}} \end{aligned}\right\} \quad i = 1, \dots, r \, . \tag{1.8}$$

iii.) Solve the LP:

$$\begin{array}{c} \text{minimize } \{\sum_{i=1}^n (c_i - v^T t_i) x_i\} \\ s.t. \;\; a_1 x_1 + \ldots + a_n x_n = b, \; x_i \ge 0, \; i = 1, \dots, n, \end{array} \tag{1.9}$$

where $v^T = (v_1, \dots, v_r)$ and $t_1, \dots, t_r$ are the columns of the matrix T. Note that the LP (1.9) can be solved by any method which provides us with a primal-dual feasible basis

$B = \{a_k, k \in I_B\}$, where $I_B = \{i_1, \ldots, i_m\}$, e.g. by the interior point method of Lustig, Marsten and Shanno, which is well suitable especially for problems of large sizes (see [4]). Include B into $\hat{B}$ in such a way that $\hat{B}$ consists now of those vectors, which trace out B from A, T_B from T in the 0th-block, and the initially selected consecutive pairs in the other blocks.

Step 1. Calculation of the corresponding basic solution:
Determine the sets S, Q ; then, determine the components of the corresponding basic solution as follows

$$\begin{aligned} x_B &:= B^{-1}b_0, \ \text{where } b_0^T := (b^T, (z_{ij_i}, i \in S)) \\ \lambda_{ij_i} &= \frac{z_{ij_i+1} - T_{iB}x_B}{z_{ij_i+1} - z_{ij_i}} \\ \lambda_{ij_i+1} &= \frac{T_{iB}x_B - z_{ij_i}}{z_{ij_i+1} - z_{ij_i}}, \qquad i = 1, \ldots, r. \end{aligned} \tag{1.10}$$

Step 2. Test for primal feasibility:
If $x_B \geq 0, \lambda_{ij_i} \geq 0, \lambda_{ij_i+1} \geq 0, \ i = 1, \ldots, r$, then STOP. The basis is primal feasible, hence also optimal. Otherwise, choose any basic component which is negative and let the corresponding vector leave the basis. Go to step 3.

Step 3.
Update all those nonbasic columns from (1.5) which may enter the basis, and compute the corresponding reduced costs $\bar{c}_p, \bar{c}_{ij_i}$ (cf. step 4 in [6]).
(Note that to perform an iteration of the DTM one does not need to price all nonbasic vectors in the last r blocks of the matrix of the equality constraints (1.5), but just those which, when they enter, do not violate the special structure of the dual feasible basis: from each of the last r blocks either one or two vectors are in the basis.)

Step 4.
Determine that vector which enters the basis by taking the minimum of the fractions of the reduced costs and the corresponding entries of the leaving variable in the tableau (cf.step 5 in [6]). Go to Step 1.

□

Remark 1.1

We briefly show that $\hat{B}$ constructed in Step 0 gives us a dual feasible basis for problem (1.4). The nonsingularity of $\hat{B}$ is trivial. To show the *dual feasibility* of $\hat{B}$, let y be the dual vector corresponding to the optimal basis B for problem (1.9), i.e. y_B is any solution of the equation

$$y_B^T B = c_B^T - v^T T_B, \tag{1.11}$$

where c_B is that part of c which corresponds to B. The optimality creterion reads as follows:

$$\begin{aligned} y_B^T a_i &= c_i - v^T t_i, \quad i \in I_B \\ y_B^T a_i &\leq c_i - v^T t_i, \quad i \notin I_B, \end{aligned}$$

which implies:

$$v^T t_i + y_B^T a_i \le c_i, \quad i = 1, \dots, n. \tag{1.12}$$

Inequalities (1.12), equalities (1.8) and Lemma 2.1 in [6] imply the dual feasibility of $\hat{B}$.

□

In the next section we explain in detail the construction of the auxiliary matrix (which is used in IDTM for updating of the working basis in product form), give the respective updating formulas and compute the increase in the value of the dual objective function during one iteration used in our algorithm to perform the largest step toward the optimum.

2. Transformation Formulas

In this section, first we present simple transformation formulas to obtain the new data of problem (1.3) from the old data, when passing from the old dual feasible basis $\hat{B}$ to the new one $\hat{B}_1$ by one of the following column exchanges:

(i) deleting the column of $-z_{qj_q}$ $(-z_{qj_q+1})$, $q \in Q$, and including a column from the 0th block;

(ii) deleting the column of $-z_{qj_q}$ $(-z_{qj_q+1})$, $q \in Q$, and including a column of $-z_{ij_i-1}$ $(-z_{ij_i+1})$, $i \in S$;

(iii) deleting a column of $-z_{qj_q}$ $(-z_{qj_q+1})$ and including a column of $-z_{qj_q+2}$ $(-z_{qj_q-1})$, $q \in Q$;

(iv) deleting a column from the 0th block and including a column of $-z_{ij_i+1}$ $(-z_{ij_i-1})$, $i \in S$;

(v) deleting a column from the 0th block and including a nonbasic column from the same block.

In what follows we frequently use the following

Lemma 1:

Let A be the following $(m \times m)$ matrix:

$$A = \begin{pmatrix} A_{11} & 0 & A_{12} \\ a_{i1} \dots a_{ij-1} & a_{ij} & a_{ij+1} \quad \dots a_{im} \\ A_{21} & 0 & A_{22} \end{pmatrix},$$

where $a_{ij} \neq 0,\ 1 \leq i, j \leq m,$ and $A_{11}, A_{12}, A_{21}, A_{22}$ are $(i-1)(j-1),\ (i-1)(m-j),$ $(m-i)(j-1),\ (m-i)(m-j)$ matrices, respectively. Let B be the matrix obtained from the matrix A by deleting the ith row and the jth column. Then, the inverse B^{-1} is obtained from the inverse A^{-1} by deleting the jth row and the ith column.

□

Now, we consider all five cases separately.

2.1. Case (i): deleting the column of $-z_{qj_q}$ $(-z_{qj_q+1}),\ q \in Q,$ and including a nonbasic column from the 0th block.

The (old) working basis B has the following form (cf. (1.7)):

$$B = \begin{pmatrix} A_B \\ T_{SB} \end{pmatrix}, \tag{2.1.1}$$

where $T_{SB} = \{T_{iB},\ i \in S)\}$. Let

$$\begin{array}{l} I_B = \{i_1, \ldots, i_m, i_{m+1}, \ldots, i_{m+s}\} \\ S = \{h_1, \ldots, h_s\},\ 0 \leq s \leq r. \end{array} \tag{2.1.2}$$

If in the (old) dual feasible basis $\hat{B}$ we exchange the column containing $-z_{qj_q}$ $(-z_{qj_q+1}),\ q \in Q$, for that column from the 0th block whose subscript is p (that is the column intersecting A at a_p), then we pass to the (new) working basis B_1. It has the form:

$$B_1 = \begin{pmatrix} A_{B_1} \\ T_{S_1B_1} \end{pmatrix}, \tag{2.1.3}$$

where

$$I_{B_1} = I_B \bigcup \{p\}, \quad S_1 = S \bigcup \{q\}. \tag{2.1.4}$$

The new working basis B_1 arises from the old one in such a way that we extend it by an additional column and row, where the subscripts of the columns $\{({a_{i_j}}^T, t^T_{S_1 i_j})^T \mid i_j \in I_{B_1}\}$ and those of the rows $\{T_{h_jB} \mid h_j \in S_1\}$ admit increasing orders. For example, if the the additional row and column are the last ones, the new working basis B_1 has the structure:

$$B_1 = \begin{pmatrix} A_B & a_p \\ T_{SB} & t_{Sp} \\ T_{qB} & t_{qp} \end{pmatrix}. \tag{2.1.5}$$

To perform one iteration of our algorithm we need to compute the inverse B_1^{-1}, given B^{-1}. This is easily done, provided that we are able to construct an *auxiliary* matrix $\tilde{B}$ (with a known inverse) such that the new working basis B_1 will be obtained from this auxiliary matrix by a column-exchange. Then, the inverse of the new working basis is obtained from the inverse of the specially constructed auxiliary matrix in product form. We distinguish the following two cases:

case (a): the leaving vector is the column of $-z_{qj_q}$
case (b): the leaving vector is the column of $-z_{qj_q+1}$,
and introduce the notations

$$\tilde{z}_q = \begin{cases} z_{qj_q+1}, \text{ in case (a)} \\ z_{qj_q}, \text{ in case (b)} \end{cases}$$
$$\bar{z}_q = \begin{cases} z_{qj_q+1} - z_{qj_q}, \text{ in case (a)} \\ z_{qj_q} - z_{qj_q+1}, \text{ in case (b)} \end{cases} \tag{2.1.6}$$
$$\tilde{\lambda}_q = \begin{cases} \lambda_{qj_q}, \text{ in case (a)} \\ \lambda_{qj_q+1}, \text{ in case (b).} \end{cases}$$

Now, we construct the auxiliary matrix $\tilde{B}$. For that, we replace the new column $(a_p^T, t_{S_1p}^T)^T$ in B_1 (given by (2.1.3)) by a column in which all elements are 0, except for $\tilde{z}_q$, the latter being placed in the position of t_{qp}. In the example (2.1.5) the auxiliary matrix has the form:

$$\tilde{B} = \begin{pmatrix} A_B & 0 \\ T_{SB} & 0 \\ T_{qB} & \tilde{z}_q \end{pmatrix}. \tag{2.1.7}$$

Let us subdivide the set of subscripts I_B into two parts L_p and U_p as follows:

$$\begin{array}{ll} I_B = L_p \bigcup U_p, & \text{where} \\ L_p := \{I_B |\, i_j < p\}, & U_p := \{I_B |\, i_j > p\}. \end{array} \tag{2.1.8}$$

Then, we have
$$I_{B_1} = L_p \bigcup \{p\} \bigcup U_p.$$

Let E_{first}^p, E_{last}^p be the matrices consisting of the (m+s+1)-component unit vectors with subscripts in L_p and U_p, respectively, and let us introduce the matrix

$$\tilde{F} = (E_{first}^p, d_p^{(1)}, E_{last}^p), \tag{2.1.9}$$

where the (m+s+1)-component vector $d_p^{(1)}$ is placed in the position of the entering column. It is computed as follows:

$$d_p^{(1)} = \tilde{B}^{-1} \begin{pmatrix} a_p \\ t_{S_1p} \end{pmatrix}. \tag{2.1.10}$$

We have that

$$B_1 = \tilde{B}\tilde{F}. \tag{2.1.11}$$

Thus, the auxiliary matrix we have been looking for is constructed. It follows that

$$B_1{}^{-1} = \tilde{F}^{-1}\tilde{B}^{-1}. \tag{2.1.12}$$

It remains to compute $\tilde{F}^{-1}$ and $\tilde{B}^{-1}$. Using Lemma 1, we obtain that $\tilde{B}^{-1}$ is obtained from B^{-1} if we extend it by one additional row and column: the additional row enters in the position of $p \in I_{B_1}$, and the additional column enters in the position of $q \in S_1$; the additional row consists of the elements in $(-\frac{T_{qB}B^{-1}}{\bar{z}_q})$ and $\frac{1}{\bar{z}_q}$, the latter being placed in the position of the additional column in $\tilde{B}^{-1}$; the additional column has elements all equal to 0, with the exception of the new element $\frac{1}{\bar{z}_q}$. In our example (2.1.7), $\tilde{B}^{-1}$ has the form:

$$\tilde{B}^{-1} = \begin{pmatrix} B^{-1} & 0 \\ -\frac{T_{qB}B^{-1}}{\bar{z}_q} & \frac{1}{\bar{z}_q} \end{pmatrix} . \tag{2.1.13}$$

Now, using (2.1.10), we can compute $d_p^{(1)}$; the resulting vector consists of the components of the vector d_p and one additional element $\frac{t_{qp}-T_{qB}d_p}{\bar{z}_q}$, the latter being placed in the position of p of the new row in $\tilde{B}^{-1}$:

$$d_p^{(1)} = \begin{pmatrix} [d_p]^p_{first} \\ \frac{t_{qp}-T_{qB}d_p}{\bar{z}_q} \\ [d_p]^p_{last} \end{pmatrix} \leftarrow p\text{th component}, \quad \text{where} \quad d_p = B^{-1}\begin{pmatrix} a_p \\ t_{Sp} \end{pmatrix} . \tag{2.1.14}$$

The vectors $[d_p]^p_{first}$, $[d_p]^p_{last}$ are obtained from the (m+s)-component vector d_p if we subdivide it into two parts: the *first* part contains those components of d_p which have subscripts in L_p, and the *last* part contains those elements which have subscripts in U_p. Having computed the $(m + s + 1)$-component vector $d_p^{(1)}$, we can compute $\tilde{F}^{-1}$:

$$\tilde{F}^{-1} = \begin{pmatrix} E^p_{first} & \begin{matrix} -\frac{d_p\bar{z}_q}{t_{qp}-T_{qB}d_p} \\ \frac{\bar{z}_q}{t_{qp}-T_{qB}d_p} \end{matrix} & E^p_{last} \end{pmatrix} . \tag{2.1.15}$$

Finally, using (2.1.12), we compute the matrix B_1^{-1} which consists of the matrix $\bar{B}^{-1}$ given below, extended by an additional row and column in the same manner as it was done to the matrix $\tilde{B}^{-1}$. The additonal row (placed in the position of p in B_1^{-1}) consists of the elements in the row $-\frac{T_{qB}B^{-1}}{t_{qp}-T_{qB}d_p}$ and of the element $\frac{1}{t_{qp}-T_{qB}d_p}$, the latter being placed in the position of the additional column. The additional column (placed in the position of q in B_1^{-1}) consists of the elements in $-\frac{d_p}{t_{qp}-T_{qB}d_p}$, except for the element $\frac{1}{t_{qp}-T_{qB}d_p}$. The matrix $\bar{B}^{-1}$ is computed as follows:

$$\bar{B}^{-1} = B^{-1} + \frac{T_{qB}d_p}{t_{qp} - T_{qB}d_p}B^{-1} .$$

In case of example (2.1.5) the inverse of the new working basis B_1 has the form:

$$B_1^{-1} = \begin{pmatrix} \bar{B}^{-1} & -\frac{d_p}{t_{qp}-T_{qB}d_p} \\ -\frac{T_{qB}B^{-1}}{t_{qp}-T_{qB}d_p} & \frac{1}{t_{qp}-T_{qB}d_p} \end{pmatrix} . \tag{2.1.16}$$

Let $\tilde{x}_B$ and x_{B_1} designate the solutions of the following systems of equations:

$$\tilde{B}\tilde{x}_B = b_1, \tag{2.1.17}$$

$$B_1 x_{B_1} = b_1, \quad \text{where} \quad b_1 := \begin{pmatrix} b \\ z_{ij_i}\ (i \in S_1) \end{pmatrix} . \tag{2.1.18}$$

It is easy to see that the vector $\tilde{x}_B$ in (2.1.17) consists of the components of the old basic solution x_B and one additonal component $\tilde{\lambda}_q$ which is placed in the position of $p \in I_{B_1}$:

$$\tilde{x}_B = \begin{pmatrix} [x_B]^p_{first} \\ \tilde{\lambda}_q \\ [x_B]^p_{last} \end{pmatrix} \leftarrow p\text{th component} , \tag{2.1.19}$$

where $[x_B]^p_{first}$ and $[x_B]^p_{last}$ are obtained from the vector x_B , if we subdivide it into two parts corresponding to the subdivision of the subscript set $I_B = L_p \bigcup U_p$. Then, using (2.1.18) and (2.1.12), we obtain that

$$x_{B_1} = B_1^{-1} b_1 = \tilde{F}^{-1}\tilde{B}^{-1} b_1 = \tilde{F}^{-1}\tilde{x}_B. \tag{2.1.20}$$

If we substitute (2.1.15) into (2.1.20), then we obtain

$$x_{B_1} = \begin{pmatrix} [x_B - x_q^{(1)} d_p]^p_{first} \\ x_q^{(1)} \\ [x_B - x_q^{(1)} d_p]^p_{last} \end{pmatrix}, \quad \text{where} \quad x_q^{(1)} := \tilde{\lambda}_q \frac{\bar{z}_q}{t_{qp} - T_{qB} d_p}. \tag{2.1.21}$$

Here $\tilde{\lambda}_q$, $\bar{z}_q$ and d_p are defined in (2.1.6), and (2.1.14), respectively; x_B is the old basic solution which is defined in (1.10), and the subdivision of $x_B - x_q^{(1)} d_p$ is again performed according to the subdivision of the set I_B into L_p and U_p. As regards the new basic $\lambda-$components, we easily find that

$$\begin{cases} \lambda^{(1)}_{ij_i+1} = \lambda_{ij_i+1} + x_q^{(1)} \frac{t_{ip} - T_{iB} d_p}{z_{ij_i+1} - z_{ij_i}} \\ \lambda^{(1)}_{ij_i} = 1 - \lambda^{(1)}_{ij_i+1} \end{cases} \qquad (i \in Q_1), \tag{2.1.22}$$

where

$$Q_1 := Q \backslash q . \tag{2.1.23}$$

To define the new *dual* vector $y_{B_1} := \{\ y_{B_1 i} \mid i \in \{1, \ldots, m\} \bigcup S_1\ \}$ we have to solve the equation

$$\begin{array}{l} (y_{B_1})^T B_1 = (\hat{c}_{B_1})^T, \quad \text{where} \\ (\hat{c}_{B_1})^T := c^T_{B_1} - (v^{(1)})^T T_{B_1}, \quad v^{(1)} := \{ v_i^{(1)} | \ i \in Q_1 \}, \end{array} \tag{2.1.24}$$

and c_{B_1}, T_{B_1} are those parts of c and T, respectively, which correspond to the basic subscripts of B_1. Now, we have that

$$\hat{c}_{B_1} = \begin{pmatrix} [\hat{c}_B + v_q T_{qB}^T]^p_{first} \\ \hat{c}_p + v_q t_{qp} \\ [\hat{c}_B + v_q T_{qB}^T]^p_{last} \end{pmatrix} \leftarrow p\text{th component}, \tag{2.1.25}$$

where

$$\hat{c}_B = c_B - \sum_{h \in Q} v_h t_{hB},\ \hat{c}_p := c_p - \sum_{h \in Q} v_h t_{hp}. \tag{2.1.26}$$

Here $v_h, h \in Q$, are the components of the old dual vector v and $\hat{c}_B + v_q T_{qB}^T$ is subdivided into two parts corresponding to the subdivision of $I_B = L_p \bigcup U_p$. Let us introduce the following notations:

$$\begin{array}{l} \bar{c}_p := \hat{c}_B^T d_p - \hat{c}_p, \\ K_{qp} = \begin{cases} \bar{c}_p/(t_{qp} - T_{qB} d_p), & \text{in case (a)}, \\ \bar{c}_p/(T_{qB} d_p - t_{qp}), & \text{in case (b)}. \end{cases} \end{array} \tag{2.1.27}$$

Then, using the previously obtained inverse B_1^{-1}, we obtain from (2.1.24):

$$y_{B_1} = \begin{pmatrix} [y_B + K_{qp}(T_{qB}B^{-1})^T]^q_{first} \\ v_q - K_{qp} \\ [y_B + K_{qp}(T_{qB}B^{-1})^T]^q_{last} \end{pmatrix} \leftarrow q\text{th component}, \tag{2.1.28}$$

where the subdivision of the (m+s)-component vector $y_B + K_{qp}(T_{qB}B^{-1})^T$ into two parts is performed corresponding to the subdivision of the set of the subscripts $\bar{S} := \{1, \dots, m\} \bigcup S_1$ into L_q and U_q with respect to the position of the subscript q:

$$\begin{array}{l} \bar{S} = L_q \bigcup U_q, \\ L_q := \{j \in \bar{S} | j < q\},\ \ U_q := \{j \in \bar{S} | j > q\}. \end{array}$$

The other dual variables $\{v_i^{(1)}, w_i^{(1)}, i \in Q_1\}$ remain unchanged.

Finally, given the old value $V = b_0^T y_B + \sum_{i \in Q} w_i)$ we compute the new dual objective function value V_1:

$$V_1 = b_1^T y_{B_1} + \sum_{i \in Q_1} w_i^{(1)}, \tag{2.1.29}$$

where b_1 are given in (2.1.18). Using (2.1.28), we obtain

$$\Delta V := V_1 - V = K_{qp}(T_{qB} x_B - \tilde{z}_q), \tag{2.1.30}$$

where $\tilde{z}_q$ and K_{qp} are defined in (2.1.6) and (2.1.27), respectively.

Summary of Case (i):

Suppose a column of $-z_{qj_q}$ $(-z_{qj_q+1}), q \in Q$, leaves the basis and a nonbasic column from the 0th block, the column of a_p, enters the basis. Then,

1.) the components of the new primal solution, corresponding to B_1, and the $\lambda-$components are given by (2.1.21) and (2.1.22), respectively;

2.) the components of the new dual vector corresponding to B_1 are given by (2.1.28), whereas all other components of the new dual vector remain unchanged;

3.) the increase ΔV in the value of the dual objective function (which is used in Section 3 to determine the largest step toward the optimum) is given by (2.1.30).

$\square$

Remark 2.1.1:

In case of a minimization problem the dual method produces a nondecreasing sequence of objective function values. Thus, (2.1.27) and (2.1.30) imply

$$\frac{\bar{c}_p}{t_{qp} - T_{qB}d_p}(T_{qB}x_B - \tilde{z}_q) \geq 0. \tag{2.1.31}$$

This has the following consequence:
in case (a) $\lambda_{qj_q} < 0$, and therefore (using (1.10))

$$T_{qB}x_B - z_{qj_q+1} > 0. \tag{2.1.32}$$

Furthermore, if the column of a_p enters the basis and $\bar{c}_p \leq 0$, then we have to restrict ourselves to the negative denominator in the expression (2.1.31), i.e., the *entering* column of a_p should be chosen in such a way that

$$t_{qp} - T_{qB}d_p < 0; \tag{2.1.33}$$

in case (b) $\lambda_{qj_q+1} < 0$, and therefore (using (1.10))

$$T_{qB}x_B - z_{qj_q} < 0\,. \tag{2.1.34}$$

Then, using similar considerations as in case (a) we obtain that the *entering* column of a_p has to be chosen in such a way that

$$t_{qp} - T_{qB}d_p > 0. \tag{2.1.35}$$

$\square$

Remark 2.1.2:

To perform the *largest step* toward the optimum we proceed in the following manner:
1.) For each $q \in Q$ such that $\lambda_{qj_q} < 0$ (or $\lambda_{qj_q+1} < 0$) determine a nonbasic subscript $p(q) \in \{1, 2, \ldots, n\} \backslash I_B$ at which the following minimum is attained

$$m(q) := \min_{t_{qp} - T_{qB}d_p < 0} \left\{ \frac{\bar{c}_p \bar{z}_q}{t_{qp} - T_{qB}d_p} \;\middle|\; \frac{\bar{z}_q}{t_{qp} - T_{qB}d_p} < 0 \right\},$$

where $\bar{z}_q$ is as in (2.1.6) and $d_p, \bar{c}_p$ are as in (2.1.14), (2.1.27).

2.) Then, that pair of *leaving* and *entering* vectors which produces the *largest increase* in the dual objective function value is given by the column of $-z_{\hat{q}}$ and the $\hat{p}$th column of the 0th block, where $\hat{p} := p(\hat{q})$. The subscript $\hat{q}$ is determined as follows

$$\text{(2.1.36)} \qquad \begin{aligned} \hat{q} := \;& arg\max\nolimits_{q\in Q}\{(T_{qB}x_B - \tilde{z}_q)K_{qp(q)}\} = \\ & arg\max\nolimits_{q\in Q}\{\tfrac{T_{qB}x_B-\tilde{z}_q}{\bar{z}_q} \; m(q) \mid \; \tfrac{T_{qB}x_B-\tilde{z}_q}{\bar{z}_q} > 0\ \}, \end{aligned}$$

where $\tilde{z}_q$ is given in (2.1.6).

Note: The considerations of Remarks 2.1.1 and 2.1.2 are used in Section 3 for the description of Step 3 of our algorithm. According to Remark 2.1.1, the pair (q,p) has to satisfy the inequalities (2.1.32), (2.1.33), if the column of z_{qj_q} is the leaving one, and (2.1.34), (2.1.35), if the column of z_{qj_q+1} is the leaving one. Hence, in Steps 3.1 and 3.2 in Section 3, the pair of subscripts (q,j_q) (or (q,j_q+1)) varies in $Q^{(1)}$ (or $Q^{(2)}$), $Q^{(1)}$, $Q^{(2)} \subset Q$), and the nonbasic subscript p varies in $P_q^{(1)}$ (or $P_q^{(2)}$) (cf. (3.1), (3.7) and (3.3), (3.8), respectively).

□

2.2. Case (ii): delete a column of $-z_{qj_q}$ $(-z_{qj_q+1})$, $q \in Q$, and include a column of $-z_{ij_i-1}$ $(-z_{ij_i+1})$, $i \in S$.

In this case the new working basis B_1 is obtained from the old working basis B by deleting the row T_{iB} and including the row T_{qB}. The new sets Q_1 and S_1 are obtained as follows:

$$\text{(2.2.1)} \qquad \begin{aligned} Q_1 &= (Q\backslash\{q\})\bigcup\{i\} \\ S_1 &= (S\backslash\{i\})\bigcup\{q\}\,. \end{aligned}$$

The new working basis B_1 has the form:

$$\text{(2.2.2)} \qquad B_1 = \begin{pmatrix} A_B \\ T_{S_1B} \end{pmatrix}.$$

First we compute the new *primal* and *dual* vectors corresponding to the working basis B_1. We solve equations (2.1.18) and (2.1.24) with the new sets Q_1 and S_1. We distinguish the following two cases:

case (a): the entering vector is the column of $-z_{ij_i-1}$

case (b): the entering vector is the column of $-z_{ij_i+1}$,

and introduce the notations

$$\text{(2.2.3)} \qquad \begin{aligned} \bar{z}_i &= \begin{cases} z_{ij_i} - z_{ij_i-1}, & \text{in case (a)} \\ z_{ij_i} - z_{ij_i+1}, & \text{in case (b)} \end{cases} \\ \tilde{\lambda}_i^{(1)} &= \begin{cases} \lambda_{ij_i}^{(1)}, & \text{in case (a)} \\ \lambda_{ij_i+1}^{(1)}, & \text{in case (b)} \end{cases} \\ \tilde{c}_i &= \begin{cases} c_{ij_i-1} - c_{ij_i}, & \text{in case (a)} \\ c_{ij_i+1} - c_{ij_i}, & \text{in case (b).} \end{cases} \end{aligned}$$

To compute the new basic solution we proceed similarly as we did in the previous section and construct an *auxiliary* matrix $\tilde{B}$. For that first we introduce an *intermediate* matrix $\tilde{B}_1$, supplementing B_1 by one additional column and row as follows:

$$\tilde{B}_1 = \begin{pmatrix} A_B & 0 \\ [T_{S_1B}]^i_{first} & 0 \\ T_{iB} & \bar{z}_i \\ [T_{S_1B}]^i_{last} & 0 \end{pmatrix} = \begin{pmatrix} [B_1]^i_{first} & 0 \\ T_{iB} & \bar{z}_i \\ [B_1]^i_{last} & 0 \end{pmatrix}. \tag{2.2.4}$$

Here the subdivision of T_{S_1B} (B_1, respectively) into $[T_{S_1B}]^i_{first}$ and $[T_{S_1B}]^i_{last}$ ($[B_1]^i_{first}$ and $[B_1]^i_{last}$, respectively) corresponds to the subdivision of the set of subscripts S_1 into the disjoint sets L_i and U_i with respect to the position i in $S_1 \bigcup \{i\}$. Thus

$$\begin{aligned} &S_1 = L_i \bigcup U_i, \\ &L_i := \{j \in S_1 \mid j < i\} \;\; U_i := \{j \in S_1 \mid j > i\}, \end{aligned} \tag{2.2.5}$$

and

$$[B_1]^i_{first} := \begin{pmatrix} A_B \\ [T_{S_1B}]^i_{first} \end{pmatrix}, \qquad [B_1]^i_{last} := [T_{S_1B}]^i_{last}.$$

Now, let $\tilde{x}_{B_1}, \tilde{y}_{B_1}$ be the solutions of the equations

$$\tilde{B}_1 \tilde{x}_{B_1} = \tilde{b}_1, \qquad \tilde{y}_{B_1} \tilde{B}_1 = \tilde{c}_{B_1}, \tag{2.2.6}$$

where

$$\tilde{b}_1 = \begin{pmatrix} b \\ z_{lj_l} \; (l \in L_i) \\ z_{ij_i} \\ z_{uj_u} \; (u \in U_i) \end{pmatrix} = \begin{pmatrix} [b_1]^i_{first} \\ z_{ij_i} \\ [b_1]^i_{last} \end{pmatrix}, \qquad \tilde{c}_{B_1} = (\hat{c}_B + v_q T_{qB} \;,\; \tilde{c}_i \;) \;. \tag{2.2.7}$$

The vectors $b_1, \hat{c}_B$ and $\tilde{c}_i$ are given in (2.1.18), (2.1.26) and (2.2.3), respectively. The vector b_1 is subdivided in accordance with the subdivision of B_1. Now, we easily see that

$$\tilde{x}_{B_1} = \begin{pmatrix} x_{B_1} \\ \tilde{\lambda}^{(1)}_i \end{pmatrix}, \quad \tilde{y}_{B_1} = \begin{pmatrix} [y_{B_1}]^i_{first} \\ v^{(1)}_i \\ [y_{B_1}]^i_{last} \end{pmatrix}, \tag{2.2.8}$$

where the subdivision of the vector y_{B_1} follows the subdivision rule of b_1 and B_1. Thus, to obtain the new primal and dual vectors corresponding to B_1, we need to compute $\tilde{B}_1^{-1}$. Similarly as in Section 2.1, we obtain $\tilde{B}_1^{-1}$ in product form, provided that we are able to construct an *auxiliary* matrix $\tilde{B}$ (with a known inverse) which is obtained from $\tilde{B}_1$ by a simple column exchange. We observe that

$$\tilde{B}_1 = \tilde{B}\tilde{F} \;, \tag{2.2.9}$$

where $\tilde{B}$ is obtained from the old working basis B, if we supplement it by one additional row and column as follows:

$$\tilde{B} = \begin{pmatrix} A_B & 0 \\ [T_{SB}]^q_{first} & 0 \\ T_{qB} & \bar{z}_q \\ [T_{SB}]^q_{last} & 0 \end{pmatrix} = \begin{pmatrix} [B]^q_{first} & 0 \\ T_{qB} & \bar{z}_q \\ [B]^q_{last} & 0 \end{pmatrix}, \tag{2.2.10}$$

and

$$\tilde{F} = \begin{pmatrix} E_{(m+s)} & d_i \\ 0 & -\frac{T_{qB}d_i}{\bar{z}_q} \end{pmatrix}, \quad \text{where} \quad d_i = B^{-1}\hat{z}_i\,. \tag{2.2.11}$$

The value of $\bar{z}_q$ is defined in (2.1.6), and the $(m+s)$-component vector $\hat{z}_i$ is obtained from the (m+s+1)-component last column of $\tilde{B}_1$, if we delete that component which corresponds to the row T_{qB}; this component is equal to 0. The subdivision of T_{SB} (B) into $[T_{SB}]^q_{first}$ and $[T_{SB}]^q_{last}$ ($[B]^q_{first}$ and $[B]^q_{last}$) is performed corresponding to the subdivision of S with respect to the position of q $(q \notin S)$. Thus

$$\begin{array}{l} S = L_q \bigcup U_q\ , \text{ where} \\ L_q := \{j \in S \mid j < q\},\ \ U_q := \{j \in S \mid j > q\}\,. \end{array} \tag{2.2.12}$$

Now, proceeding in a similar way as we did in Section 2.1 and using (2.1.21), we obtain

$$\tilde{x}_{B_1} = \begin{pmatrix} x_B - \tilde{\lambda}^{(1)}_i d_i \\ \tilde{\lambda}^{(1)}_i \end{pmatrix}, \qquad \tilde{\lambda}^{(1)}_i = \tilde{\lambda}_q \frac{\bar{z}_q}{-T_{qB}d_i}\,, \tag{2.2.13}$$

where $\tilde{\lambda}_q$ and d_i are given in (2.1.6) and (2.2.11), respectively. Furthermore, from (2.1.28) we obtain that

$$\tilde{y}_{B_1} = \begin{pmatrix} [y_B + K_{qi}(T_{qB}B^{-1})^T]^q_{first} \\ v_q - K_{qi} \\ [y_B + K_{qi}(T_{qB}B^{-1})^T]^q_{last} \end{pmatrix}, \tag{2.2.14}$$

where the subdivision of the vector $y_B + K_{qi}(T_{qB}B^{-1})^T$ follows the rule given in (2.2.12). The number K_{qi} is computed as follows:

if the column of $-z_{qj_q+1}$ $(-z_{qj_q})$ leaves the basis, then

$$\begin{array}{ll} \text{in case (a):} & K_{qi} = \frac{\hat{c}_B d_i - \tilde{c}_i}{\bar{z}_i T_{qB} u_i} \quad \left(-\frac{\hat{c}_B d_i - \tilde{c}_i}{\bar{z}_i T_{qB} u_i}\right) \\ \text{in case (b):} & K_{qi} = \frac{\hat{c}_B d_i - \tilde{c}_i}{\bar{z}_i T_{qB} u_i} \quad \left(-\frac{\hat{c}_B d_i - \tilde{c}_i}{\bar{z}_i T_{qB} u_i}\right). \end{array} \tag{2.2.15}$$

Here $u_i := d_i/\bar{z}_i$ and $\hat{c}_B$, $\tilde{c}_i$ are defined in (2.1.26), (2.2.3), respectively. Finally, the new value of the dual objective function is:

$$V_1 = K_{qi}(T_{qB} - \tilde{z}_q)\,, \tag{2.2.17}$$

where $\tilde{z}_q$ is given in (2.1.6). Now, using the (intermediate) solutions given in (2.2.13), (2.2.14), we obtain the new primal basic solution and dual vector as follows:

$$x_{B_1k} = \tilde{x}_{B_1k}, \qquad k \in I_B$$

(2.2.18)
$$\tilde{\lambda}_i^{(1)} = \begin{cases} \frac{\lambda_{qj_q}(z_{qj_q+1}-z_{qj_q})}{-T_{qB}d_i}, & \text{if } -z_{qj_q} \text{ leaves the basis} \\ \frac{\lambda_{qj_q+1}(z_{qj_q+1}-z_{qj_q})}{T_{qB}d_i}, & \text{if } -z_{qj_q+1} \text{ leaves the basis} \end{cases}$$

(2.2.19)
$$\begin{array}{l} y_{B_1l} = \tilde{y}_{B_1l} \quad, \ l \in \{1,\dots,m\} \bigcup S_1 \quad (l \neq i) \\ v_i^{(1)} = \tilde{c}_i/\bar{z}_i\,, \qquad w_i^{(1)} = c_{ij_i} - z_{ij_i}v_i^{(1)}. \end{array}$$

The other components of the new primal and dual vectors remain the same as in the previous step.

Summary of Case (ii):

Suppose a column of $-z_{qj_q}(-z_{qj_q+1}), q \in Q$, leaves the basis and either (a) a column of $-z_{ij_i+1}$, or (b) a column of $-z_{ij_i-1}$, $i \in S$, enters the basis.

1.) The new primal basic solution is computed as follows:

(2.2.20)
$$x_{B_1k} = \begin{cases} x_{Bk} - \lambda_{ij_i}^{(1)}d_{ik}, & \text{in case (a)} \\ x_{Bk} - \lambda_{ij_i+1}^{(1)}d_{ik}, & \text{in case (b)} \end{cases} \qquad k \in I_{B_1}$$

$$\lambda_{lj_l}^{(1)} = \frac{z_{lj_l+1} - T_{lB}x_{B_1}}{z_{lj_l+1} - z_{lj_l}}\,, \quad \lambda_{lj_l+1}^{(1)} = 1 - \lambda_{lj_l}^{(1)}, \qquad l \in Q_1 \backslash \{i\},$$

where d_{ik} are the components of the vector d_i given in (2.2.11),

(2.2.21)
$$\begin{array}{l} \lambda_{ij_i}^{(1)} = \begin{cases} \tilde{\lambda}_i^{(1)}, & \text{in case (a)} \\ 1 - \tilde{\lambda}_i^{(1)}, & \text{in case (b)} \end{cases} \\ \lambda_{ij_i+1}^{(1)} = 1 - \lambda_{ij_i}^{(1)}, \end{array}$$

and $\tilde{\lambda}_i^{(1)}$ is given in (2.2.18).

2.) The new dual vector corresponding to the working basis B_1 is computed as follows

$$y_{B_1l}^T = y_{Bl}^T + K_{qi}T_{qB}B^{-1}\,, \quad l \in \{1,\dots,m\} \bigcup S_1\,,$$

where K_{qi} is given by (2.2.15). The components $v_i^{(1)}, w_i^{(1)}$ are defined in (2.2.19) and the other components of $v^{(1)}, w^{(1)}$ remain the same as in the previous step.

3.) The increase ΔV in the value of the dual objective function is:

(2.2.22)
$$\Delta V = K_{qi}(T_{qB}x_B - \tilde{z}_q)\,,$$

where $\tilde{z}_q$ is given in (2.1.6); (2.2.22) is used in Section 3 for the determination of the largest increase in the objective function value.

Remark 2.2.1:

To perform the *largest step* toward the optimal value we proceed as follows:

1.) For each q such that $\lambda_{qj_q} < 0$ or $\lambda_{qj_q+1} < 0$ let

$$[(i,j)](q) \in \{(i,j) \mid i \in S,\ j = j_i \pm 1,\ 0 < j_i \le k_i\}$$

denotes that *nonbasic subscript* at which the following minimum is attained:

$$\begin{aligned} m(q) &:= \min\{m^+,\ m^-\}\ ,\ \text{where} \\ m^+ &= \min_{i\in S} \{\ \tfrac{\bar{c}_{ij_i+1}\tilde{z}_q}{\bar{z}_i T_{qB} u_i} \ \mid\ \tfrac{\tilde{z}_q}{\bar{z}_i T_{qB} u_i} \le 0\ \}, \\ m^- &= \min_{i\in S} \{\ \tfrac{\bar{c}_{ij_i-1}\tilde{z}_q}{\bar{z}_i T_{qB} u_i} \ \mid\ \tfrac{\tilde{z}_q}{\bar{z}_i T_{qB} u_i} \le 0\ \}. \end{aligned}$$

Here $u_i := d_i/\bar{z}_i$, $\bar{c}_{ij_i\pm 1} := \hat{c}_B d_i - \tilde{c}_i$ and d_i, $\bar{z}_i$ are defined in (2.2.11), (2.2.3), respectively.

2.) The pair of *leaving* and *entering* vectors which produces the largest increase in the dual objective function is given by the columns of $-z_{qj_q}$ and $-z_{ij}$, where $(i,j) := [(i,j)](q)$, and the subscript q is determined by the equation:

$$q = arg\max_{t\in Q} \{\frac{T_{tB}x_B - \tilde{z}_t}{\bar{z}_t} m(t) \mid \frac{T_{tB}x_B - \tilde{z}_t}{\bar{z}_t} \ge 0\}\ , \tag{2.2.23}$$

where $\bar{z}_t$ and $\tilde{z}_t$, $t \in Q$, are defined in (2.1.6).

□

The considerations of Remark 2.2.1 are used in Section 3 for the description of Step 3 of our Algorithm, where the the expression for ΔV in (2.2.22) is used to determine the “optimal” pair of the leaving and entering vectors (cf. the last two lines in (3.1) and (3.7), or the last two lines in (3.3) and (3.8)).

2.3. Case (iii): delete the column of $-z_{qj_q}(-z_{qj_q+1})$ and include the column of $-z_{qj_q+2}(-z_{qj_q-1})$, $\quad q \in Q$.

The column exchange in the qth block of the dual feasible basis $\hat{B}$ changes only those components of the primal basic solution, which correspond to the qth block. To obtain the new *primal* basic solution we need only to compute the new values of $\{\lambda^{(1)}_{qj_q}, \lambda^{(1)}_{qj_q+1}\}$, the other basic components of $\{\lambda^{(1)}_{ij}\}$, as well as the basic components in x_{B_1}, remain the same. To obtain the new *dual* vector y_{B_1}, we need to compute only those components of

$\{v_i^{(1)}, w_i^{(1)}, i \in Q_1\}$, which correspond to the qth block, while the other components of $v^{(1)}$ and $w^{(1)}$ remain unchanged. Note that $Q_1 = Q$. Let us introduce the notations

$$\begin{aligned} D_r &:= [z_{qj_q}, z_{qj_q+1}, z_{qj_q+2}; c_{q.}](z_{qj_q+2} - z_{qj_q}) \\ D_l &:= [z_{qj_q-1}, z_{qj_q}, z_{qj_q+1}; c_{q.}](z_{qj_q+1} - z_{qj_q-1}), \end{aligned} \tag{2.3.1}$$

where $q \in Q$ and $[z_{qj_q}, z_{qj_q+1}, z_{qj_q+2}; c_{q.}]$, $[z_{qj_q-1}, z_{qj_q}, z_{qj_q+1}; c_{q.}]$ denote the second order divided differences of the sequence c_{qj}, $j = 0, \ldots, k_q + 1$, (cf. (1.2)). We distinguish the following two cases:

case (a): a column of $-z_{qj_q}$ leaves the basis and a column of $-z_{qj_q+2}$ enters the basis

case (b): a column of $-z_{qj_q+1}$ leaves a basis and the column of $-z_{qj_q-1}$ enters the basis.

Using (1.10) we can compute the new basic components $\{\lambda_{qj_q}^{(1)}, \lambda_{qj_q+1}^{(1)}\}$ as follows:

$$\begin{aligned} \lambda_{qj_q+1}^{(1)} &= \begin{cases} \frac{-(z_{qj_q+1} - T_{qB}x_B)}{(z_{qj_q+2} - z_{qj_q+1})}, & \text{in case (a)} \\ \frac{(T_{qB}x_B - z_{qj_q-1})}{(z_{qj_q} - z_{qj_q-1})}, & \text{in case (b)} \end{cases} \\ \lambda_{qj_q}^{(1)} &= 1 - \lambda_{qj_q+1}^{(1)}. \end{aligned} \tag{2.3.2}$$

All the other components of $\{\lambda_{ij}^{(1)}\}$ remain unchanged. From (1.8) we derive

$$\begin{aligned} v_q &= - \frac{c_{qj_q+1} - c_{qj_q}}{z_{qj_q+1} - z_{qj_q}} = -[z_{qj_q}, z_{qj_q+1}; c_{q.}] \\ v_q^{(1)} &= \begin{cases} -[z_{qj_q+1}, z_{qj_1+2}; c_{q.}], & \text{in case (a)} \\ -[z_{qj_q-1}, z_{qj_q}; c_{q.}], & \text{in case (b).} \end{cases} \end{aligned}$$

Using the notations (2.3.1) we obtain

$$v_q^{(1)} = \begin{cases} v_q - D_r, & \text{in case (a)} \\ v_q + D_l, & \text{in case (b).} \end{cases} \tag{2.3.3}$$

Again, using the equations (1.8), we have:

$$\begin{aligned} w_q &= c_{qj_q+1} + z_{qj_q+1} v_q \\ w_q^{(1)} &= c_{qj_q+1} + z_{qj_q+1} v_q^{(1)}. \end{aligned}$$

If we subtract the second equation from the first one we obtain

$$w_q^{(1)} = \begin{cases} w_q - D_r z_{qj_q+1}, & \text{in case (a)} \\ w_q + D_l z_{qj_q}, & \text{in case (b).} \end{cases} \tag{2.3.4}$$

All the other components of $v^{(1)}, w^{(1)}$ remain unchanged. Now, using (1.11) and (2.3.3), the new dual vector can be computed as follows:

$$y_{B_1}^T = \begin{cases} y_B^T + D_r T_{qB} B^{-1}, & \text{in case (a)} \\ y_B - D_l T_{qB} B^{-1}, & \text{in case (b).} \end{cases} \tag{2.3.5}$$

Finally, for the new value of the dual objective function we obtain

$$(2.3.6) \qquad V_1 := \begin{cases} V - D_r(z_{qj_q+1} - T_{qB}x_B), \text{ in case (a)} \\ V - D_l(T_{qB}x_B - z_{qj_q}), \text{ in case (b)}, \end{cases}$$

where

$$V = b^T y_B + \sum_{i \in Q} w_i.$$

If we use the expressions for the reduced costs (cf. [6]):

$$\bar{c}_{qj_q-1} := -D_l(z_{qj_q} - z_{qj_q-1})$$
$$\bar{c}_{qj_q+2} := -D_r(z_{qj_q+2} - z_{qj_q+1}),$$

then the new value of the dual objective function can be computed as follows:

$$(2.3.7) \qquad V_1 = \begin{cases} V + \bar{c}_{qj_q+2}\frac{z_{qj_q+1} - T_{qB}x_B}{z_{qj_q+2} - z_{qj_q+1}} = V - \bar{c}_{qj_q+2}\lambda^{(1)}_{qj_q+1}, \text{ in case (a)} \\ \\ V + \bar{c}_{qj_q-1}\frac{-(z_{qj_q} - T_{qB}x_B)}{z_{qj_q} - z_{qj_q-1}} = V - \bar{c}_{qj_q-1}\lambda^{(1)}_{qj_q}, \text{ in case (b)}. \end{cases}$$

Summary of Case (iii):

1.) The components of the *primal* basic solution $\{x_{B_1};\ \lambda^{(1)}_{ij_i}, i \in S;\ \lambda^{(1)}_{lj_l}, l \in Q\}$ remain the same in the next step, with the exception of the components $\lambda^{(1)}_{qj_q}$, $\lambda^{(1)}_{qj_q}$, which correspond to the *qth* block and are given by (2.3.2).

2.) Those components of the *dual* vector y_{B_1} which correspond to the new working basis B_1 are given by (2.3.5), and those ones which correspond to qth block are given by (2.3.3), (2.3.4). All the other components remain the same.

3.) The new value of the dual objective function V_1 is presented in (2.3.7).

Remark 2.3.1:
In case (a): the qth-component of v decreases during the iteration; furthermore,
if $z_{qj_q+1} > 0$, then w_q decreases too, but if $z_{qj_q+1} < 0$, then w_q increases.
In case (b): the qth-components of v increases during the iteration; furthermore,
if $z_{qj_q} > 0$, , then w_q increases too; if $z_{qj_q} < 0$, then w_q decreases.

In fact, by the convexity of the discrete function c_i. we have that D_r, D_l, given in (2.3.1), are positive, and this implies:

$$v^{(1)}_q < v_q, \text{ in case (a)}$$
$$v^{(1)}_q > v_q, \text{ in case (b)}.$$

The assertion concerning w_q can be proved similarly.

Note: this above mentioned property of the components of the dual vector is helpful to select the "optimal" initial pairs of the basic vectors in the last r blocks of the matrix (1.5).

□

Remark 2.3.2:
To perform the *largest step* toward the optimal value we proceed similarly as in Remarks 2.1.1 and 2.1.2. The expression (2.3.7) is used for the description of the Algorithm in Section 3 to determine the "optimal" pair of the leaving and entering vectors: compare (2.3.7) with $m_1(q), M_{\mathbf{a}}$ in (3.1), (3.7), and $m_5(q), M_{\mathbf{b}}$ in (3.3), (3.8).

□

2.4. Case (iv): delete a column from the 0th block and include a column of $-z_{ij_i+1}(-z_{ij_i-1})$, $i \in S$.

In this case we delete that column which intersect A at a_p and include

case (a): the column of $-z_{ij_i+1}$, $i \in S$

case (b): the column of $-z_{ij_i-1}$, $i \in S$.

The new working basis B_1 has one column and one row less than the old working basis B, and we have an additional block in $\hat{B}_1$ containing a consecutive pair of vectors, i.e.:

(2.4.1)
$$I_{B_1} = I_B \backslash \{p\}, \quad S_1 = S \backslash \{i\}, \quad Q_1 = Q \bigcup \{i\}.$$

Similarly as we have done in Section 2.1, we construct an *auxiliary* matrix $\tilde{B}$ (the inverse of which can be computed in product form), so that the inverse of B_1 can be derived from the inverse of $\tilde{B}$ by simple operations. Let us introduce the following notations:

(2.4.2)
$$\begin{aligned} \bar{z}_i &= \begin{cases} z_{ij_i} - z_{ij_i+1}, & \text{in case } (a) \\ z_{ij_i} - z_{ij_i-1}, & \text{in case } (b) \end{cases} \\ \tilde{z}_i^T &= (\overbrace{0,\ldots,0}^{m+(i-1)}, \bar{z}_i, \overbrace{0,\ldots,0}^{(s-i)}) \\ \bar{c}_i &= \begin{cases} c_{ij_i+1} - c_{ij_i}, & \text{in case } (a) \\ c_{ij_i-1} - c_{ij_i}, & \text{in case } (b). \end{cases} \end{aligned}$$

We construct the auxiliary matrix $\tilde{B}$ by replacing the column $\tilde{z}_i$ (given in (2.4.2)) for the column $(a_p^T,\ t_{lp}^T\ (l \in S))^T$ in the old working basis B. The only non-zero element of $\tilde{z}_i$ is placed in the position of the element $t_{ip}, i \in S$. The corresponding row T_{iB} will leave B, since the corresponding block i will contain a consecutive pair of vectors and, consequently, it does not anymore belong to the 0th block in the new dual feasible basis $\hat{B}_1$. For example, if the column of a_p and the row T_{iB} are the last ones, our auxiliary matrix $\tilde{B}$ has the following form:

$$\tilde{B} = \begin{pmatrix} B_1 & 0 \\ T_{iB_1} & \bar{z}_i \end{pmatrix}. \tag{2.4.3}$$

It is easy to see that we obtain the new working basis B_1 from our auxiliary matrix $\tilde{B}$ by simply deleting the ith row and the new column $\tilde{z}_i$. Then, using Lemma 1, we obtain the inverse of the new working basis B_1 by excluding the pth row and ith column from $\tilde{B}^{-1}$. Now, let us compute $\tilde{B}^{-1}$. For that we subdivide the set of the basic subscripts I_B, given in (2.1.2), as follows:

$$I_B = L_p \bigcup \{p\} \bigcup U_p, \quad L_p := \{i_j \in I_B \mid i_j < p\}, \quad U_p := \{i_j \in I_B \mid i_j > p\}. \tag{2.4.4}$$

Let E^p_{first}, E^p_{last} be the matrices consisting of the (m+s)-component unit vectors with subscripts in L_p, U_p, respectively, and introduce the following matrix:

$$F = (E^p_{first},\ d,\ E^p_{last}). \tag{2.4.5}$$

The (m+s)-component vector d is placed in the position p of the outgoing column in B. If u_i designates the ith column of B^{-1}, then we have the equation

$$d = B^{-1}\tilde{z}_i = u_i \bar{z}_i. \tag{2.4.6}$$

It follows that

$$\tilde{B} = BF, \quad \tilde{B}^{-1} = F^{-1}B^{-1},$$

where

$$F^{-1} = \begin{pmatrix} & -u_{i1}/u_{ip} & \\ & -u_{i2}/u_{ip} & \\ & \vdots & \\ E^p_{first} & 1/(u_{ip}\bar{z}_i) & E^p_{last} \\ & \vdots & \\ & -u_{im+s}/u_{ip} & \end{pmatrix} \tag{2.4.7}$$

and u_{ik} $(k \in I_B)$ denotes the kth component of the vector u_i.
Let $\tilde{x}_B$ be the solution of $\tilde{B}\tilde{x}_B = b_0$, where b_0 is defined in (1.10). Then we have that

$$\tilde{x}_B = F^{-1}B^{-1}b_0 = F^{-1}x_B, \tag{2.4.8}$$

where x_B is the old basic solution. Now, using (2.4.7), we obtain

$$\begin{aligned} \tilde{x}_{Bk} &= x_{Bk} - (u_{ik}/u_{ip})x_{Bp} \ , \text{ for } k \in I_{B_1} \\ \tilde{x}_{Bp} &= \frac{1}{u_{ip}\bar{z}_i} x_{Bp} \ , \end{aligned} \tag{2.4.9}$$

where x_{Bj} , $\tilde{x}_{Bj}(j \in I_B)$ denote the jth component of the corresponding vectors.

Now, we observe that the components of the primal basic solution corresponding to B_1 (the solution of the equation (2.1.18)), as well as those components of the primal basic solution which correspond to the last r blocks and are going to be changed during the present iteration, can immediately be determined through the components of our auxiliary solution as follows:

$$
\begin{array}{l}
x_{B_1 k} = \tilde{x}_{Bk}, \text{ for all } \ k \in I_{B_1} \\
\lambda^{(1)}_{ij_i \pm 1} = \tilde{x}_{Bp} \ , \qquad \lambda^{(1)}_{ij_i} = 1 - \lambda^{(1)}_{ij_i \pm 1} \\
\text{(here } + \text{ or } - \text{ is chosen depending on the incoming vector).}
\end{array} \tag{2.4.10}
$$

The other components of the primal basic solution remain the same.

To obtain the new *dual* vector, we have to solve equation (2.1.24) with the new working basis B_1 and corresponding I_{B_1}, S_1 and Q_1. First we compute the (auxiliary) solution $\tilde{y}_{B_1}$ of the following equation:

$$
\tilde{y}_B^T \tilde{B} = \tilde{c}_B^T, \qquad \text{where} \qquad \tilde{c}_B = \begin{pmatrix} [\hat{c}_B]^p_{first} \\ \bar{c}_i \\ [\hat{c}_B]^p_{last}) \end{pmatrix} . \tag{2.4.11}
$$

We obtain

$$
\tilde{y}_B^T = \tilde{c}_B^T \tilde{B}^{-1} = \tilde{c}_B^T F^{-1} B^{-1}. \tag{2.4.12}
$$

Now, using the expression for F^{-1} given in (2.4.7), we derive

$$
\tilde{y}_B^T = \begin{pmatrix} [\hat{c}_B]^p_{first} \\ \hat{c}_p + \frac{\bar{c}_i - \hat{c}_B^T d}{d_p} \\ [\hat{c}_B]^p_{last} \end{pmatrix}^T B^{-1} \ , \tag{2.4.13}
$$

where the vector d is defined in (2.4.6) and d_p is the pth component of the vector d. Since we have

$$
y_B^T = \begin{pmatrix} [\hat{c}_B]^p_{first} \\ \hat{c}_p \\ [\hat{c}_B]^p_{last} \end{pmatrix}^T B^{-1} \ ,
$$

it follows that

$$
\tilde{y}_B^T = y_B^T + \frac{\bar{c}_i - \hat{c}_B^T d}{d_p} g_p \ , \tag{2.4.14}
$$

where g_p denotes the pth row of B^{-1}. Now we can easily compute our new dual vector by the use of $\tilde{y}_B$. Taking into account the shape of the matrix $\tilde{B}^{-1}$, and using (2.4.12), we obtain that

$$
\begin{array}{l}
y_{B_1 k} = \tilde{y}_{Bk} - \bar{c}_i \tilde{g}_{pk} \ , \ \text{for all } \ k \in \{1, \ldots, m\} \bigcup S_1 \ \ (k \neq i) \\
v_i^{(1)} = \tilde{y}_{Bi} \\
w_i^{(1)} = c_{ij_i} + z_{ij_i} v_i^{(1)},
\end{array} \tag{2.4.15}
$$

where $\tilde{g}_{pk}$ is the kth element of the pth row of $\tilde{B}^{-1}$, and $\bar{c}_i$ is given in (2.4.2). Using (2.4.6), we can compute the components of the pth row of $\tilde{B}^{-1}$:

$$\tilde{g}_{pk} = g_{pk}/(u_{ip}\bar{z}_i), \qquad \text{for all} \quad k \in \{1,\dots,m\} \bigcup S_1 \; (k \neq i). \tag{2.4.16}$$

The increase of the value of the dual objective function value is:

$$\begin{aligned} \Delta V \; &:= V_1 - V = (b_1^T y_{B_1} + z_{ij_i} v_i^{(1)}) - b_0^T y_B = \\ &= b_0(\tilde{y}_B - y_B) = \tfrac{\bar{c}_i - \hat{c}_B^T d}{d_p} g_p b_0 = \tfrac{\bar{c}_i - \hat{c}_B^T d}{d_p} x_{Bp} \,, \end{aligned} \tag{2.4.17}$$

where b_0, b_1, $\hat{c}_B^T$ and d are given in (1.10), (2.1.18), (2.1.26) and (2.4.6), respectively. Now, using the following expression for the reduced costs

$$\bar{c}_{ij_i \pm 1} = \hat{c}_B^T d - \bar{c}_i \quad (+ \text{ or } - \text{ is chosen depending on the incoming vector}), \tag{2.4.18}$$

and substituting (2.4.6) for d, we can write:

$$\Delta V = -\frac{\bar{c}_{ij_i \pm 1}}{\bar{z}_i u_{ip}} x_{Bp} \,. \tag{2.4.19}$$

Summary of Case(iv):

Suppose the column of a_p from the 0th block leaves the basis, and we include
case (a): a column of $-z_{ij_i+1}$, $i \in S$
case (b): a column of $-z_{ij_i-1}$, $i \in S$.
Then we have the following results:

1.) The components of the new primal solution, corresponding to the new working basis B_1 and to the ith block of the new basis $\hat{B}_1$, are given by (2.4.10). The rest of the basic components of $\{\lambda_{ij}\}$ are:

$$\lambda_{kj_k}^{(1)} = \frac{z_{kj_k+1} - T_{kB_1} x_{B_1}}{z_{kj_k+1} - z_{kj_k}}, \qquad \lambda_{kj_k+1}^{(1)} = 1 - \lambda_{kj_k}^{(1)} \qquad (k \in Q_1 \backslash \{i\}).$$

2.) The components of the dual vector, corresponding to B_1 and to the ith block, are given by (2.4.15). The rest of the components of the new dual vector remain the same.
3.) The increase in the value of the dual objective function, which is used to determine the largest step toward the optimum, is given by (2.4.19).

□

Remark 2.4.1:
To perform the *largest step* toward the optimal value we proceed similarly as it is described in Remarks 2.1.1 and 2.1.2. The expression (2.4.19) is used in the description of Step 3 of the Algorithm in Section 3 to determine the "optimal" pair of the leaving and entering vectors: compare (2.4.19) with $m_{10}(k)$, $m_{11}(k)$ and $M_{\mathbf{c}}$ in (3.5) and (3.9).

□

2.5. Case (v): delete a column from the 0th block and include a nonbasic column from the same block.

The considerations are similar to those in the previous case, hence it is enough to summarize the results.

Summary of Case (v):

Suppose the column of a_ℓ from the 0th block leaves the basis and a nonbasic column of a_p from the 0th block enters the basis. Then we have the following results:

1.) The new basic components out of $x_1, \ldots, x_n$ are:

$$\begin{cases} x_{B_1k} = x_{Bk} - \frac{d_{pk}}{d_{p\ell}} x_{B\ell}\,, \text{ for } k \in I_{B_1} \ \ (k \neq p) \\ x_{B_1p} = \frac{x_{B\ell}}{d_{p\ell}} \end{cases}$$

where

(2.5.1) $$I_{B_1} = (I_B \bigcup \{p\}) \backslash \{\ell\}, \ \ d_p = B^{-1} \begin{pmatrix} a_p \\ t_{S_p} \end{pmatrix} .$$

2.) The new basic λ-components are:

(2.5.2) $$\lambda^{(1)}_{ij_i+1} = \frac{T_{iB_1} x_{B_1} - z_{ij_i}}{z_{ij_i+1} - 1_{ij_i}}\,, \ \ \lambda^{(1)}_{ij_i} = 1 - \lambda^{(1)}_{ij_i+1}, \ i \in Q_1,$$

where $Q_1 := Q$.

3.) The new dual vector y_{B_1} is:

(2.5.3) $$\begin{cases} y_{B_1k} = y_{Bk}\,, \quad k \in I_{B_1}, \quad k \neq p \\ y_{B_1p} = y_{B\ell} - \frac{\bar{c}_p}{d_{p\ell}} g_\ell, \end{cases}$$

where $\bar{c}_p$ denotes the reduced cost (cf. (2.1.27)) and g_ℓ denotes the ℓth row of the inverse B^{-1}; the other components of the dual solution remain the same.

4.) The increase in the value of the dual objective function is:

(2.5.4) $$\Delta V = -\frac{\bar{c}_p}{d_{p\ell}} x_{B\ell}.$$

□

Remark 2.5.1.:

To perform the *largest step* toward the optimum we proceed similarly as mentioned in Remarks 2.1.1 and 2.1.2. The expression (2.5.4) is used in Section 3 in the description of Step 3 of the Algorithm to determine the "optimal" pair of the leaving and entering vectors: compare (2.5.4) with the second line in (3.5) and (3.9).

□

3. An Algorithm

First we introduce the following notations:

$$\bar{I}_B := \{1,\ldots,n\}\backslash I_B$$
$$\hat{I}_B := \{k \in I_B \mid x_{Bk} < 0\}.$$

and for $q \in Q,\ 0 \le j_q \le k_q$:

$$\begin{aligned} Q^{(1)} &:= \{(q,j_q) \mid \lambda_{qj_q} < 0\} \\ Q^{(2)} &:= \{(q,j_q+1) \mid \lambda_{qj_q+1} < 0\} \\ P_q^{(1)} &:= \{p \in \bar{I}_B \mid t_{qp} - T_{qB}d_p < 0\} \\ P_q^{(2)} &:= \{p \in \bar{I}_B \mid T_{qB}d_p - t_{qp} < 0\} \\ S_q^{(1)} &:= \{i \in S \mid T_{qB}u_i > 0\} \\ S_q^{(2)} &:= \{i \in S \mid T_{qB}u_i < 0\}. \end{aligned}$$

Here $d_p = B^{-1}\begin{pmatrix} a_p \\ t_{Sp} \end{pmatrix}$, and u_i is the ith column of B^{-1}. The subscripts of the nonbasic columns of the matrix (1.5) are:

$$\bar{I}_B\ ,\ \{(q,j) \mid j \ne j_q, j_q+1,\ q \in Q\}\ ,\ \{(i,j) \mid j \ne j_i\ ,\ i \in S\}.$$

In the Algorithm we distinguish three cases: the leaving vector has the subscript

case (a): $(q,j_q) \in Q^{(1)}$

case (b): $(q,j_q+1) \in Q^{(2)}$

case (c): $k \in \hat{I}_B$.

In case (a) we introduce the notations:

$$\begin{aligned} m_{\mathbf{a}}(q) &= \min\{m_1(q),m_2(q),m_3(q),m_4(q)\}\ , \qquad \text{where} \\ m_1(q) &:= \tfrac{-\bar{c}_{qj_q+2}}{z_{qj_q+2}-z_{qj_q+1}} \\ m_2(q) &:= \min_{p\in P_q^{(1)}} \frac{\bar{c}_p}{t_{qp}-T_{qB}d_p} \\ m_3(q) &:= \min_{i\in S_q^{(1)},\ j_i>0} \left\{-\frac{\bar{c}_{ij_i-1}}{(z_{ij_i}-z_{ij_i-1})T_{qB}u_i}\right\} \\ m_4(q) &:= \min_{i\in S_q^{(2)},\ j_i<k_i} \left\{\frac{\bar{c}_{ij_i+1}}{(z_{ij_i+1}-z_{ij_i})T_{qB}u_i}\right\}\ . \end{aligned} \tag{3.1}$$

Moreover, we denote by $\alpha(q)$ that element of the set of nonbasic subscripts for which

$$\alpha(q) = arg\min\{m_1(q),m_2(q),m_3(q),m_4(q)\}\ , \qquad \text{i.e.,}$$

$$\alpha(q) = \begin{cases} (q,j_q+2) & \text{if } m_{\mathbf{a}}(q) = m_1(q) \\ \{p \mid p \in P_q^{(1)}\}\ , & \text{if } m_{\mathbf{a}}(q) = m_2(q) \\ \{(i,j_i-1) \mid i \in S_q^{(1)}\}, & \text{if } m_{\mathbf{a}}(q) = m_3(q) \\ \{(i,j_i+1) \mid i \in S_q^{(2)}\}, & \text{if } m_{\mathbf{a}}(q) = m_4(q) \end{cases} \tag{3.2}$$

($\alpha(q)$ gives the subscript of the entering vector which is determined according to the DM).

In case (b) we define:

$$
(3.3) \qquad \begin{array}{l}
m_{\mathbf{b}}(q) = \min\{m_5(q), m_6(q), m_7(q), m_8(q)\} \\
m_5(q) := -\bar{c}_{qj_q-1}/(z_{qj_q} - z_{qj_q-1}) \\
m_6(q) := \min\limits_{p \in P_q^{(2)}} \{-\bar{c}_p/(t_{qp} - T_{qB}d_p)\} \\
m_7(q) := \min\limits_{i \in S_q^{(2)},\, j_i > 0} \{\bar{c}_{ij_i-1}/[(z_{ij_i} - z_{ij_i-1})T_{qB}u_i]\} \\
m_8(q) := \min\limits_{i \in S_q^{(1)},\, j_i < k_i} \{-\bar{c}_{ij_i+1}/[(z_{ij_i+1} - z_{ij_i})T_{qB}u_i]\} \,.
\end{array}
$$

Moreover, we denote by $\beta(q)$ that element of the set of nonbasic subscripts for which

$$
(3.4) \qquad \begin{array}{l}
\beta(q) = arg \min\{m_5(q), m_6(q), m_7(q), m_8(q)\} \,, \quad \text{i.e.,} \\
\beta(q) = \begin{cases}
(q, j_q - 1) \,, & \text{if} \quad m_{\mathbf{b}}(q) = m_5(q) \\
\{p \mid p \in P_q^{(2)}\}, & \text{if} \quad m_{\mathbf{b}}(q) = m_6(q) \\
\{(i, j_i - 1) \mid i \in S_q^{(2)}\}, & \text{if} \quad m_{\mathbf{b}}(q) = m_7(q) \\
\{(i, j_i + 1) \mid i \in S_q^{(1)}\}, & \text{if} \quad m_{\mathbf{b}}(q) = m_8(q)
\end{cases}
\end{array}
$$

($\beta(q)$ gives the subscript of the entering vector which is determined according to the DM).

Finally, in case (c) we define:

$$
(3.5) \qquad \begin{array}{l}
m_{\mathbf{c}}(k) = \min\{m_9(q), m_{10}(q), m_{11}(q)\} \\
m_9(k) := \min\limits_{p \in \bar{I}_B \,,\, d_{pk} < 0} \{\bar{c}_p/d_{pk}\} \\
m_{10}(k) := \min\limits_{i \in S \,,\, j_i > 0, u_{ik} < 0} \{\bar{c}_{ij_i-1}/[(z_{ij_i} - z_{ij_i-1})u_{ik}]\} \\
m_{11}(k) := \min\limits_{i \in S \,,\, j_i < k_i, u_i(k) > 0} \{-\bar{c}_{ij_i+1}/[(z_{ij_i+1} - z_{ij_i})u_{ik}]\}.
\end{array}
$$

Here x_{Bk}, d_{pk}, u_{ik} denote the kth components of the corresponding vectors.
Now, we denote by $\gamma(k)$ that element of the set of nonbasic subscripts for which

$$
(3.6) \qquad \begin{array}{l}
\gamma(k) = arg \min\{m_9(q), m_{10}(q), m_{11}(q)\} \,, \quad \text{i.e.,} \\
\gamma(k) = \begin{cases}
\{p \mid p \in \bar{I}_B \,,\ d_{pk} < 0\}, & \text{if} \quad m_{\mathbf{c}}(k) = m_9(k) \\
\{\ (i, j_i - 1) \mid \ i \in S \,,\ u_{ik} < 0\}, & \text{if} \quad m_{\mathbf{c}}(k) = m_{10}(k) \\
\{\ (i, j_i + 1) \mid \ i \in S \,,\ u_{ik} > 0\}, & \text{if} \quad m_{\mathbf{c}}(k) = m_{11}(k)
\end{cases}
\end{array}
$$

($\gamma(k)$ gives the subscript of the entering vector which is determined according to the DM).

The considerations of the previous section suggest the following

Algorithm IDTM:

Step 1. Determine an initial dual feasible basis B_0 and the corresponding basic solution: carry out the algorithm described in Section 1.

Step 2. Test for primal feasibility:
If $x_B \geq 0, \lambda_{ij_i} \geq 0, \lambda_{ij_i+1} \geq 0, i = 1, \ldots, r$, then STOP. Otherwise, go to Step 3.

Step 3. Simultaneous selection of the *leaving* and *entering* vectors:

3.1. For all $\{q | q \in Q, q \in Q^{(1)}\}$ compute

$$M_{\mathbf{a}} = \max\{(T_{qB}x_B - z_{qj_q+1})m_{\mathbf{a}}(q)\}, \tag{3.7}$$

where $m_{\mathbf{a}}(q)$ is computed according to (3.1).

3.2. For all $\{q | q \in Q, q \in Q^{(2)}\}$ compute

$$M_{\mathbf{b}} = \max\{(z_{qj_q} - T_{qB}x_B)m_{\mathbf{b}}(q)\}, \tag{3.8}$$

where $m_{\mathbf{b}}(q)$ is computed according to (3.3).

3.3. For all $\{k | k \in \hat{I}_B\}$ compute

$$M_{\mathbf{c}} = \max\{-x_{Bk}\, m_{\mathbf{c}}(k)\}, \tag{3.9}$$

where $m_{\mathbf{c}}(k)$ is computed according to (3.5).

3.4. Compute

$$M = \max\{M_{\mathbf{a}},\ M_{\mathbf{b}},\ M_{\mathbf{c}}\}.$$

3.5. Finally, select the *leaving* and the *entering* vectors as described below.

3.5.(a) If M is equal to $M_{\mathbf{a}}$ and the minimum in (3.1) is attained at q, then the pair of *leaving* and *entering* vectors which produces the *largest step* toward the optimal value is either given by the vectors of $-z_{qj_q}$, $-z_{\alpha(q)}$, or by the vector of $-z_{qj_q}$ and the column from the 0th block with the subscript $\alpha(q)$; $\alpha(q)$ is determined in (3.2). The *new basic solution* is defined

– if $m_{\mathbf{a}}(q) = m_1(q)$: as described in the Summary of Case (iii);

– if $m_{\mathbf{a}}(q) = m_2(q)$: as described in the Summary of Case (i);

– if $m_{\mathbf{a}}(q) = m_3(q)$ or $m_{(a)}(q) = m_4(q)$: as described in the Summary of Case (ii).

3.5.(b) If M is equal to $M_{\mathbf{b}}$ and the minimum in (3.2) is attained at q, then the *pair of leaving and entering vectors* which produces the *largest step* towards the optimal value is given by the vectors of $-z_{qj_q+1}$, $-z_{\beta(q)}$, or by the vector of $-z_{qj_q+1}$, and the column from the 0th block with subscript $\beta(q)$; $\beta(q)$ is determined in (3.4). The *new basic solution* is defined

– if $m_{\mathbf{b}}(q) = m_5(q)$: as described in the Summary of Case (iii);

– if $m_{\mathbf{b}}(q) = m_6(q)$: as described in the Summary of Case(i);

– if $m_{\mathbf{b}}(q) = m_7(q)$ or $m_{\mathbf{b}}(q) = m_8(q)$: as described in the Summary of Case (ii).

3.5.(c) If M is equal to $M_{\mathbf{c}}$ and the minimum in (3.3) is attained at k, then the pair of *leaving* and *entering* vectors which produces the *largest step* toward the optimal value is given either by the columns of a_k and $a_{\gamma(k)}$ from the 0th block, or by the column of a_k from the 0th block and the column of $-z_{\gamma(k)}$ out of the last r blocks; the subscript $\gamma(k)$ is given in (3.6). The *new basic solution* is defined

– if $m_{\mathbf{c}}(k) = m_9(k)$: as described in the Summary of Case (v);

– if $m_{\mathbf{c}}(k) = m_{10}(q)$ or $m_c(k) = m_{11}(q)$: as described in the Summary of Case (iv).

Go to **Step 2**.

□

4. Implementation

The implementation of IDTM is based on C.I.Fábián's general purpose optimization routine library called LINX. It is a tool for *interactive* solution of linear programming problems. Interactivity here means that this routine collection supports the solution of a series of LP problems, where problems can be generated successively using the experience from the solutions of the former ones.
LINX was written in *C*, so it is portable: it can be compiled without modification under MS-DOS, OS/2, and different UNIX operating systems, including ULTRIX for DEC and SOLARIS for SUN workstations. LINX consists of an LP solver "engine" and a memory handling "hull". The solver is an implementation of the two-phase revised simplex method with the product form of the inverse. Both the primal and the dual methods are implemented. The data can be scaled before optimization to decrease differences between magnitudes. Tolerances for comparing floating-point numbers are used adaptively, following the suggestions of Maros (see [5]).

Unstable transformations are avoided. Small pivots are rejected even at the price of creating small new infeasibilities. This pivot-selection rule is due to Harris (see [1]). Multiple pricing is used, so candidate vectors can also be rejected at little cost. Precision is checked from time to time. If round-off errors exceed a certain limit (or if the array describing the transformations performed that far is too long), then a reinversion process is started. It is a stabilized version of Hellerman and Rarick's Preassigned Pivot Procedure (see [2]). The original method is unstable. The version used here proved effective over 4 years of experience.

The *speed of computation* was the next objective after reliability. A CRASH procedure is available to break initial degeneracy and speed up optimization. A practical implementation of the steepest-edge method can also be used optionally. Special sort and search methods accelerate vector and pivot selection. The memory handling hull ensures that computations of former problems are available when required. If a problem is derived from a former one by

– addition or deletion of variables
– addition or deletion of constraints
– modification of the 'rim' (the right-hand-side, the objective function, or the lower-upper bounds of the variables),

then not only a starting solution is available for the new problem, but the connected basis-inverse as well. Similar ideas can be found in Zoutendijk [10]. Basis-inverses are stored in a wrap-around buffer, the size of which depends on the size of all available memory. If we have more memory, we can remember more basis-inverses, and the solution of the problem sequence will be faster. The buffer is also a stack (LIFO structure). This strategy supports the solution of a tree-structured problem set. In the simplest case, the inverses in the stack are connected to problems each of which is derived from the one preceding it. LINX was built into several optimization packages, and solved real-life problems with up to 9 000 contstraints.

The above routine collection could easily be extended to implement IDTM. In IDTM, we solve two connected problems: (1.9) and the main problem (1.4). The memory handling part helps in moving from one problem to the other. We had to modify the subroutine that inverts the basis, and carries out FTRAN and BTRAN: it is made capable to handle the varying size working basis throughout the procedure. Pricing is also modified, because in Step 3 of the DTM or IDTM we only have to price one or two vectors from each of the blocks 1,...,r. This is because a dual feasible basis may only contain at least one but at most two consecutive vectors from each of those blocks.

There are two input files and two output files, for the deterministic (A, b, c) and the stochastic (T, z, corresponding probabilities and penalties) data. They are assembled in such a way that sensitivity analysis can easily be done. The code is currently tested and reports about the performance of the IDTM will soon be available.

5. Conclusions and Future Directions

In this paper we have presented an Improved Dual Type Method (IDTM) for a linearly constrained optimization problem with a separable objective function, where the terms are continuous, convex and piecewise linear functions.
We began with a short description of the Dual Type Method (developed by A. Prékopa (1990)) and then improved on it in different ways. In Section 2 we gave the transformation formulas for updating the data (in each special case of the column exchange in the block-structured dual feasible basis). The updating formulas are summarized at the end of each case. Then, the results are used in Section 3 to create an efficient solution procedure. The implementations issues are presented in Section 4.

Problems of further interest include: 1.) how to select the initial consecutive pairs in the last r blocks in order to reduce the number of the required dual steps in the solving algorithm; 2.) how to apply the improved DTM to multistage stochastic programming problems.

References

[1] **P.M.J. Harris** (1973). Pivot selection methods, *Mathematical Programming* 5, 1–28

[2] **E. Hellerman, D.C. Rarick** (1971). Reinversion with the preassigned pivot procedure, *Mathematical Programming* 1, 195–216

[3] **C.E. Lemke** (1954). The dual method for solving the linear programming problem, *Naval Research Logistic Quarterly* 1, 36-47.

[4] **I. Lustig, R. Marsten, D. Shanno** (1991). Computational experience with a primal-dual interior point method for linear programmimg, *Linear Algebra and its Applications* 152, 191-461.

[5] **I. Maros** (1986). A general phase-I method in linear programming, *European Journal of Operational Research* 23, 64–77

[6] **A. Prékopa** (1990). Dual method for the solution of a one-stage stochastic programming problem with random RHS obeying a discrete probability distribution, *ZOR-Methods and Models of Operations Research* 34, 441-461.

[7] **A. Prékopa** (1992). A very short introduction to linear programmimg, *Rutcor Lecture Notes*, RLN 2-92.

[8] **A. Prékopa, W.Li** (1995). Solution of and bounding in a linearly constrained optimization problem with convex, polyhedral objective function, *Mathematical Programming*, to appear.

[9] **R.J. Wets** (1983). Solving stochastic programs with simple recourse, *Stochastics* 10, 219-242.

[10] **G. Zoutendijk** (1970). A Product-form algorithm using contracted transformation vectors, In: *Integer and Nonlinear Programming*, edited by J. Abadie, North-Holland, 511–523