INFORMATICA, 2000, Vol. 11, No. 4, 421–434

Computing True Shadow Prices in Linear Programming

James K. HO

Information and Decision Sciences, University of Illinois at Chicago
m/c 294, 601 South Morgan, Chicago, IL 60607, USA
e-mail: imho@uic.edu

Received: August 2000

Abstract. It is well known that in linear programming, the optimal values of the dual variables can be interpreted as shadow prices (marginal values) of the right-hand side coefficients. However, this is true only under nondegeneracy assumptions. Since real problems are often degenerate, the output from conventional LP software regarding such marginal information can be misleading. This paper surveys and generalizes known results in this topic and demonstrates how true shadow prices can be computed with or without modification to existing software.

Key words: linear programming, shadow prices, optimization software.

1. Introduction

In most elementary treatment of linear programming, such as typically found in textbooks on Management Science and Operations Research, the dual variables of an LP are interpreted as marginal values of the right-hand side coefficients. As the latter often represent resources of limited supply, such marginal values have come to be known as shadow prices. They indicate how much additional units of the corresponding resources are worth. However, this equivalence between dual variables and shadow prices holds only under the assumption of nondegeneracy. Their nonequivalence in general, while perhaps well known among specialists, is almost never discussed in textbooks for students and would be practitioners (exceptions are, e.g., Murty (1983) and Shapiro (1979)). See also Rubin and Wagner (1990) for a managerially oriented discussion.) Even commercial software for LP fails to alert users of this caveat. As a result, misleading outputs of LP models may have resulted in many inadvertent misuses of the approach. The purpose of this article is to summarize known results on this topic that have appeared in the literature, generalize them to handle any LP formulation, examine how true shadow prices can be computed with existing software, and show extensions to LP software that may be useful for the automatic generation of such prices.

2. Previous Results

For conciseness, we summarize only theoretical results in the literature, well known or otherwise, that are essential to the computation of shadow prices. A state-of-the-art survey with an extensive bibliography was given by Gal (1986). Related topics such as uniqueness of solutions and degeneracy of LP's can be found in other works in the list of references (Evans and Baker (1982), Gal (1986), Greenberg (1986), Mangasarian (1979), Pérold (1981)).

Consider the following primal-dual pair of linear programs:

$$\text{(P)}\quad \begin{array}{ll} \text{maximize} & c^{\mathrm{T}}x \\ \text{subject to} & Ax \leqslant b \\ & x \geqslant 0; \end{array}$$

$$\text{(D)}\quad \begin{array}{ll} \text{Minimize} & b^{\mathrm{T}}y \\ \text{subject to} & A^{\mathrm{T}}y \geqslant c \\ & y \geqslant 0; \end{array}$$

where A is $m \times n$, $c \in R^n$, $x \in R^n$, $b \in R^m$, $y \in R^m$.

DEFINITION 1. Denote the optimal objective value for (P), as a function of the right-hand side, by

$$v(b) = \max\{c^{\mathrm{T}}x \mid Ax \leqslant b;\ x \geqslant 0\};$$

the set of feasible solutions and the set of optimal solutions to (P) by

$$X = \{x \in R^n \mid Ax \leqslant b;\ x \geqslant 0\},$$
$$X^* = \{x \in X \mid c^{\mathrm{T}}x = v(b)\}$$

respectively; and the set of feasible solutions and the set of optimal solutions to (D) by

$$Y = \{y \in R^m \mid A^{\mathrm{T}}y \geqslant c,\ y \geqslant 0\},$$
$$Y^* = \{y \in Y \mid b^{\mathrm{T}}y = v(b)\}.$$

PROPOSITION 1 [See, e.g., Murty (1983), Rockafeller (1970)]. $v(b)$ is a non-decreasing, piecewise linear concave function.

DEFINITION 2. The set of subgradients for a concave function $f\colon R^m \to R$ is defined as

$$\partial f(b) = \{y \in R^m \mid f(b+u) \leqslant f(b) + u^{\mathrm{T}}y, \quad \text{for all } u \in R^m\}.$$

PROPOSITION 2 [See, e.g., Ch. 8 in Murty (1983)]. When $v(b)$ is finite, $\partial v(b) = Y^*$.

DEFINITION 3. The directional derivative of a function $f\colon R^m \to R$ at b in the direction of u is defined as

$$D_u f(b) = \lim_{t \to 0^+} \frac{f(b+tu) - f(b)}{t}.$$

PROPOSITION 3 [See, e.g., Gauvin (1980), Rockafeller (1970), Shapiro (1979)].

$$D_u v(b) = \min\{u^{\mathrm{T}} y \mid y \in \partial v(b)\}.$$

DEFINITION 4. For (P), the buying (or positive) shadow price of constraint i is defined as

$$p_i^+ = D_{e(i)} v(b),$$

and the selling (or negative) shadow price of constraint i is defined as

$$p_i^- = -D_{-e(i)} v(b),$$

where $e(i)$ is the ith unit vector.

In other words, the buying shadow price is the (instantaneous) rate of change in $v(b)$ for an increase in b_i and the selling shadow price is the negative of the (instantaneous) rate of change in $v(b)$ for a decrease in b_i.

PROPOSITION 4 [Gauvin (1980)].

$$\begin{aligned} p_i^+ &= \min\{y_i \mid y \in Y^*\}, \\ p_i^- &= \max\{y_i \mid y \in Y^*\}. \end{aligned}$$

DEFINITION 5. Denote the index set of the basic variables in an optimal tableau for (P) by B, the index set of the nonbasic variables by N, the index set of slack variables by S. Denote the row of reduced costs by d, the right-hand side by β, the ith row and the jth column of the tableau by $\hat{a}^i$ and $\hat{a}_j$ respectively. Let d_S and $\hat{a}_S^i$ be d and $\hat{a}^i$ restricted to S respectively. Let $T = \{i \mid \beta i = 0\}$ be the index set for the rows with degeneracy and y^* be the current dual optimal solution, i.e., $y^* = d_S$.

PROPOSITION 5 [Akgül (1984), Best (1982)]. Y^* is characterized by

$$y \in Y^* \iff y = y^* + \sum\{t_k \hat{a}_S^k \mid k \in T\}$$

for some $t \in R^{|T|}$ such that

$$d + \sum\{t_k \hat{a}^k \mid k \in T\} \geqslant 0.$$

3. The General Case

The above definition of shadow prices found in the literature (e.g., Akgül (1984), Gal (1986), Gauvin (1980), Murty (1983)) assumes an LP in the canonical form (P) such that "prices" are always nonnegative quantities. This way, $v(b)$ tends to increase or decrease with b_i. To accommodate both maximization and minimization problems with any combination of inequality as well as equality constraints we need a more general definition that allows a consistent sign convention.

DEFINITION 6a. For any LP, the incremental shadow price p_{i+} of constraint i is defined as the instantaneous rate of *improvement* in $v(b)$ with an increase in b_i; the decremental shadow price (p_{i-}) of constraint i is defined as the negative of the instantaneous rate of *improvement* in $v(b)$ with a decrease in b_i.

In this sense, a negative incremental price or a positive decremental price is the rate of deterioration of the objective value as the right-hand side is changed. Note that whether an improvement actually involves an increase or a decrease in the objective value $v(b)$ depends on the direction of optimization. For example, when minimizing it is possible to incur an increase in $v(b)$ by decreasing the right-hand side of a less-than-or-equal-to constraint. The increase in $v(b)$ is a negative improvement. This gives a negative instantaneous rate of improvement with a decrease in b_i. By definition, the decremental shadow price which is the negative of this rate comes out positive. Although it may appear somewhat cumbersome at first sight, this definition of shadow prices is necessary for consistency in the general case.

Table 1
Interpretation of shadow prices

Minimization:		
Type of constraint	Increase in $v(b)$	Decrease in $v(b)$
$\leqslant$	$p_{i-} < 0$	$p_{i+} > 0$
$\geqslant$	$p_{i+} < 0$	$p_{i-} > 0$
$=$	$p_{i+} < 0$ or $p_{i-} < 0$	$p_{i+} > 0$ or $p_{i-} > 0$;
Maximization:		
Type of constraint	Increase in $v(b)$	Decrease in $v(b)$
$\leqslant$	$p_{i+} > 0$	$p_{i-} < 0$
$\geqslant$	$p_{i-} > 0$	$p_{i+} < 0$
$=$	$p_{i+} > 0$ or $p_{i-} > 0$	$p_{i+} < 0$ or $p_{i-} < 0$;

DEFINITION 6b. Suppose LP in general has the form

$$
\text{(LP)}\qquad
\begin{array}{ll}
\text{maximize} & c^{\mathrm{T}}x \\
\text{subject to} & Lx \leqslant p \\
 & Gx \geqslant q \\
 & Ex = r \\
 & x \geqslant 0.
\end{array}
$$

Let

$$
\text{(P}'\text{)} \quad \begin{array}{ll}
\text{maximize} & c^{\mathrm{T}}x \\
\text{subject to} & Lx \leqslant p \\
& -Gx \leqslant -q \\
& Ex \leqslant r \\
& -Ex \leqslant -r \\
& x \geqslant 0;
\end{array}
$$

and

$$
\text{(D}'\text{)} \quad \begin{array}{ll}
\text{minimize} & p^{\mathrm{T}}y_L - q^{\mathrm{T}}y_G + r^{\mathrm{T}}(y_{E+} - y_{E-}) \\
\text{subject to} & L^{\mathrm{T}}y_L - G^{\mathrm{T}}y_G + E^{\mathrm{T}}(y_{E+} - y_{E-}) \geqslant c \\
& y_L,\ y_G,\ y_{E+},\ y_{E-} \geqslant 0
\end{array}
$$

be the expression of (LP) in the form of (P) and (D) in §2. Finally let the equivalent problem to (D′) be

$$
\text{(DLP)} \quad \begin{array}{ll}
\text{minimize} & p^{\mathrm{T}}y_L - q^{\mathrm{T}}y_G + r^{\mathrm{T}}y_E \\
\text{subject to} & L^{\mathrm{T}}y_L - G^{\mathrm{T}}y_G + E^{\mathrm{T}}y_E \geqslant c \\
& y_L \geqslant 0 \\
& y_G \leqslant 0 \\
& y_E \text{ unrestricted.}
\end{array}
$$

In the following proposition, Y and Y^* for (LP) is defined using (DLP). This convention is standard practice in LP implementation (see, e.g., Ho (1987), IBM, MPSX/370 (1979), Shapiro (1979)). Finally, let Y' and $Y^{'*}$ correspond to (D′). Note that y_G in (DLP) is $-y_G$ in (D′) and y_E in (DLP) is $y_{E+} - y_{E-}$ in (D′).

PROPOSITION 6. For any LP:

$$
\begin{aligned}
p_{i+} &= \min\{y_i \mid y \in Y^*\} \\
p_{i-} &= \max\{y_i \mid y \in Y^*\}.
\end{aligned}
$$

Proof. First consider the case of maximization. For less-than-or-equal-to constraints, the results follow from Proposition 4. A greater-than-or-equal-to constraints $a^i x \geqslant b_i$ can be written as $-a^i x \leqslant -b_i$ as in (P′). Applying Proposition 4 to (P′) and (D′)

$$
\begin{aligned}
p_{i+} &= \text{instantaneous rate of improvement with increase of } b_i \text{ in (LP)} \\
&= \text{instantaneous rate of improvement with decrease of } -b_i \text{ in (P}'\text{)} \\
&= -\max\{y_i \mid y \in Y'*\} \quad \text{[by applying Proposition 4 to (P}'\text{)]} \\
&= \min\{-y_i \mid y \in Y^{'*}\} \\
&= \min\{y_i \mid y \in Y^*\}. \qquad \text{[since } y_G \text{ in (DLP) is } -y_G \text{ in (D}'\text{)]}
\end{aligned}
$$

Expressing an equal-to constraint as two less-than-or-equal-to constraints as in (P′), we have

$$
\begin{aligned}
p_{i+} &= \text{instantaneous rate of improvement with increase of } b_i \text{ in (LP)} \\
&= \text{instantaneous rate of improvement with increase of } b_i \\
&\quad + \text{instantaneous rate of improvement with decrease of } -b_i \text{ in (P}'\text{)} \\
&= \min\{y_{iE+} \mid y \in Y^{'*}\} - \max\{y_{iE-} \mid y \in Y^{'*}\} \text{ [by Proposition 4 on (P}'\text{)]} \\
&= \min\{y_{iE+} \mid y \in Y^{'*}\} + \min\{-y_{iE-} \mid y \in Y^{'*}\} \\
&= \min\{y_{iE+} - y_{iE-} \mid y \in Y^{'*}\} \\
&= \min\{y_{iE} \mid y \in Y^*\}. \quad \text{[since } y_E \text{ in (DLP) is } y_{iE+} - y_{iE-} \text{ in (D}'\text{)]}
\end{aligned}
$$

The proof for p_{i-} is similar.

For minimization, the objective $\min c^{\mathrm{T}}x$ can be written as $-\max(-c^{\mathrm{T}}x)$. Both (D′) and (DLP) are as before except $-c$ replaces c. Since any rate of improvement in $\max(-c^{\mathrm{T}}x)$ is the same as that in $\min(c^{\mathrm{T}}x)$, the results follow.

Since y^*, the current dual optimal solution given by the optimal basis, belongs to Y^*, some or all (e.g., when (P) is nondegenerate) of its components may already be true shadow prices. Therefore, we need to identify such cases and then proceed to find the remaining missing shadow prices. This can be done using right-hand side ranging in LP sensitivity analysis.

DEFINITION 7. For constraint i, let r_{i+} and r_{i-} be the allowable increase and allowable decrease given by the right-hand side range analysis.

PROPOSITION 7. If $r_{i+} > 0$, then $p_{i+} = y_i^*$. If $r_{i-} > 0$, then $p_{i-} = y_i^*$.

Proof. It suffices to show for the case of (P). The general case follows with appropriate sign manipulations.

If $r_{i+} > 0$, then for small enough $\Delta b_i > 0,\ v(b+e_i\Delta b_i) = c_B B^{-1}(b+e_i\Delta b_i)$ where B is the current optimal basis, c_B are the objective coefficients of the basic variables, and e_i is the ith unit column vector. Therefore

$$\begin{aligned} p_{i+} &= \lim[\Delta b_i \to 0+]\{v(b + e_i\Delta b_i) - v(b)\}/\Delta b_i \\ &= c_B B^{-1} e_i \\ &= y_i^*. \end{aligned}$$

The proof for p_{i-} is similar.

4. Algorithms

For any shadow price p_i^+ or p_i^- that is not given by y^*, two approaches can be taken for its computation.

I. Direct Search using:

a) parametrization;
b) perturbation;

II. Constrained Dual using:

c) dual simplex;
d) implicit Y*.

a) Right-Hand Side Parametrization

By varying the right-hand side b_i parametrically, an adjacent basis (in the sense of Murty (1983)) with a different objective value is sought. If found, the dual variable y_i for this new basis provides a rate of improvement leading into this basis, which will be the same as the rate of improvement leading out of our original optimal basis. If the objective value will not change, the rate of improvement is zero. In this case, either the direction of change is infeasible or any change in b_i in that direction will not affect the objective value.

b) Right-Hand Side Perturbation

This is direct application of the definition of shadow prices. The right-hand side coefficient is altered by a suitably small amount, the LP reoptimized and the rate of change computed. See Dantzig (1981) for a more sophisticated version of this approach.

c) Constrained Dual

This follows from Proposition 6. Each missing shadow price requires the solution of (DLP) with $b^{\mathrm{T}}y = v(b)$ and y_i optimized. With conventional software, this method is too cumbersome to be practical. In enhanced software, the method below is more efficient.

d) Implicit Y^*

In this method, any missing shadow price can be solved by a subproblem implied by Propositions 5 and 6. For example, for p_{i+}, we have

$$\text{(S+)}\qquad \begin{array}{ll} \text{Minimize} & [y^* + \sum\{t_k \hat{a}_S^k \mid k \in T\}]e_i \\ \text{subject to} & \sum\{t_k \hat{a}^k \mid k \in T\} \geqslant -d. \end{array}$$

where $t \in R^{|T|}$. For finding p_{i-}, a subproblem (S_-) is obtained from (S_+) maximizing the same objective. In Section 7, efficient implementation of this approach will be discussed.

5. A Numerical Example

Before duscussing the implementation of methods to compute true shadow prices, it will help fix ideas by considering a numerical example. Consider the following LP adapted from Ho (1987), Chapter 3.

$$\begin{array}{lllllll} \text{Minimize} & 74A & +40B & +50C & +10D & \\ \text{Subject to} & 3A & +B & & +2C & \geqslant 8 \\ & 2A & +2B & & +D & \geqslant 11 \\ & 4A & & +3C & & \geqslant 10.667 \\ & A,\ B,\ C,\ D \geqslant 0 \end{array}$$

Let $S1,\ S2,\ S3$ be the slack variables and $X,\ Y,\ Z$ be the dual variables to the first, second and third constraints respectively. The optimal solution to the LP is

$A = 2.667$
$B = 0.0$
$C = 0.0$
$D = 5.667$
$S1 = 0.0$
$S2 = 0.0$
$S3 = 0.0$

with an objective value of 254.00. However, $S1$ is basic and the primal optimal solution is degenerate. Indeed, there are two alternative optimal dual (basic) solutions, namely,

$X = 0.0$
$Y = -10.0$
$Z = -13.5$

and

$X = -18.0$
$Y = -10.0$
$Z = 0.0.$

Note that the output of any commercially available LP software lists only one or the other dual optimal solution which cannot be interpreted directly as marginal values. The minimum and maximum values for the optimal dual solutions are given in Table 2.

The interpretation of these values as shadow prices are tabulated in Table 3.

The Δb_i and $\Delta v(b)$ columns are obtained from actual perturbations of the LP. The results verify Proposition 6.

Table 2
Extreme values in Y^* for example LP

Constraint	min	max
First	-18.0	0.0
Second	-10.0	-10.0
Third	-13.5	0.0

Table 3
Shadow prices for the example LP.

Constraint	Δb_i	$\Delta v(b)$	p_{i+}	p_{i-}
First	-0.1	0.0		0.0
	+0.1	+1.8	-18.0	
Second	-0.1	-1.0		-10.0
	+0.1	+1.0	-10.0	
Third	-0.1	0.0		0.0
	+0.1	+1.35	-13.5	

6. Computing with Conventional LP Codes

Given that the outputs of conventional LP codes do not provide complete information on all the true shadow prices it is of interest to see if one can still obtain such answers using only the very same codes. For this purpose, we use two popular packages: LINDO (1986) on IBM PC's and MPSX/370 (1979) on IBM mainframes.

Two different procedures are suggested for use with LINDO:

i) parametric right-hand side;

ii) perturbation.

In the parametric approach, LINDO is used to solve the LP and do the Range (Sensitivity) analysis. See Fig. 1 for the screen output at this stage. For each right-hand side that has an allowable increase of zero, the following procedure is performed. The PARA command is used with a new right-hand side value that is large enough to force a change in the objective value if possible.

Fig. 2 shows the output from PARA for the first constraint in our example. The dual variable (-18.00) corresponding to the first new objective value gives the incremental shadow price (p_{i+}) sought. In cases where the objective value does not change, the last dual variable listed should be used.

Similarly, for each right-hand side that has an allowable decrease of zero, the PARA command is used with a decrement to the right-hand side value to find the decremental shadow price. Fig. 3 shows the output from PARA for the third constraint in the example.

Note that whenever the objective changes in one case of parametrization, the LP needs to be resolved before proceeding with another case. This is necessary because LINDO starts with the last available tableau to execute the PARA option. Therefore, performing consecutive parametric analysis will not lead to desired results. This is a definite drawback of this approach with LINDO as the LP may have to be solved as many times as there are missing shadow prices.

The perturbation procedure involves using the ALTER command to change a right-hand side value, again by an appropriately small amount. Then the LP is resolved and the output is examined to determine missing shadow prices using their definitions directly. The results of this procedure for the example are listed in Table 3 in §5.

With LINDO, this approach is not as inefficient as it may seem because the last tableau available is used to start an altered LP. However additional computation must be performed to determine the rates of change. Also, the perturbations must be made independently. This means the original problem must be retrieved at each step, or another ALTER must be used to erase the previous case.

Using MPSX/370, two procedures are suggested. First, solve LP with the RANGE option. Examine the ranges for row at limit levels for correct cases of shadow prices. For all other cases, use the PARARHS option. In MPSX/370, this is based on incrementing the original right-hand side by successive multiples of a change column until a maximum increment is reached. All three parameters: the change column, the multiple, and the maximum increment must be specified in the control program. The change column is

```
: MIN 74A+40B+50C+10D
?ST
?3A+B+2C>8
?2A+2B+D>11
?4A+3C>10.666667
?END
:GO
LP OPTIMUM FOUND AT STEP 2

          OBJECTIVE FUNCTION
1)                254.000000
 VARIABLE              VALUE  REDUCED COST
        A           2.666667       .000000
        B            .000000     20.000000
        C            .000000      9.500000
        D           5.666667       .000000

      ROW  SLACK OR SURPLUS   DUAL PRICES
       2)            .000000       .000000
       3)            .000000    -10.000000
       4)            .000000    -13.500000

NO. ITERATIONS = 2

DO RANGE(SENSITIVITY) ANALYSIS?
?Y

RANGES IN WHICH THE BASIS IS UNCHANGED:

                  OBJ COEFFICIENT RANGES
  VARIABLE    CURRENT  ALLOWABLE  ALLOWABLE
                 COEF   INCREASE   DECREASE
         A  74.000000  12.666670  54.000000
         B  40.000000   INFINITY  20.000000
         C  50.000000   INFINITY   9.500000
         D  10.000000  10.000000   6.333333

       ROW    CURRENT  ALLOWABLE  ALLOWABLE
                  RHS   INCREASE   DECREASE
         2   8.000000    .000000    .000000
         3  11.000000   INFINITY   5.666667
         4  10.666670    .000000    .000000
```

Fig. 1. LINDO output for example LP.

```
: PARA
ROW:
2
NEW RHS VALUE =
8.001
      VAR   VAR   PIVOT    RHS       DUAL       OBJ
      OUT    IN    ROW     VAL     VARIABLE     VAL

 ART        SLK 4    2    8.00000   .000000   254.000
       D    SLK 3    3    16.5000  -18.0000   407.000
```

Fig. 2. LINDO PARA output for first constraint.

```
: PARA
ROW:
4
NEW RHS VALUE =
10.6
      VAR   VAR   PIVOT    RHS       DUAL       OBJ
      OUT    IN    ROW     VAL     VARIABLE     VAL

 SLK   2    SLK 4    2    10.6667  -13.5000   254.000
 ART         ART     0    10.6667   .000000   254.000
```

Fig. 3. LINDO PARA output for last constraint.

the appropriate unit vector for finding an incremental shadow price and the negative unit vector for finding a decremental shadow price. The parameters should be chosen to reduce extraneous computation and output.

Note that in MPSX/370, each PARARHS command is based on the optimal tableau of the LP and not on the tableau for the previous parametrization. Therefore, no redundant resolution of the LP is necessary as is the case with LINDO.

Since MPSX/370 is not interactive, the above procedure can become cumbersome. For the same reason, the perturbation method is deemed impractical. However, the parametric approach can be automated at the expense of extraneous computation. This second procedure is carried out by setting up in a single control program all the PARARHS runs, regardless of whether they eventually become necessary or not.

7. Computing with Enhanced LP Codes

As we have seen above, although it is possible to reconstruct all true shadow prices for an LP by repeated application of available features in conventional software, it would be much more convenient to have such information generated automatically. This will of course involve the modification of existing codes. To gain insight into the complexities

```
      VARIABLE                VALUE       REDUCED COST
             1   2.666666667E+00   0.000000000E+00
             2   0.000000000E+00   2.000000000E+00
             3   0.000000000E+00   1.400000000E+00
             4   5.666666667E+00   0.000000000E+00
           ROW              LOGICAL         DUAL VALUE
             1  -2.540000000E+02   1.000000000E+00
             2   0.000000000E+00  -1.800000000E+01
             3   0.000000000E+00  -1.000000000E+01
             4   6.357828776E-07   5.551115123E-17
 SHADOW PRICES:
            ROW        INCREMENTAL        DECREMENTAL
             2  -1.800000000E+01   0.000000000E+00
             3  -1.000000000E+01  -1.000000000E+01
             4  -1.350000241E+01   0.000000000E+00
```

Fig. 4. Output from experimental LP code.

of such attempts we have extended an experimental LP code to include the computation of true shadow prices using method d) described earlier.

Consider subproblem (S_+) in Section 4. Each column in this LP is the transpose of a degenerate row in the optimal tableau of (P). There are $n + m$ constraints, corresponding to the variables (including slacks) in (P). Note that a basic variables in the optimal tableau implies either a null constraint in (S_+) or a nonnegativity constraint on a t_k. Therefore, (S_+) effectively has n constraints in nonnegative variables $t_k, k \in T$. Computationally, it is more efficient to solve its dual. Letting J be the index set of nonbasic variables in the optimal tableau of (P), the dual to (S_+) can be written as follows.

$$(\mathrm{DS}_+) \quad \begin{array}{ll} \text{Maximize} & \sum -d_j w_j \\ \text{subject to} & \sum \hat{a}_j^k w_j \leqslant \hat{a}_{Si}^k, \quad k \in T; \\ & \quad w_j \geqslant 0, \quad j \in J. \end{array}$$

Each column in (DS+) is simply a column in the optimal tableau of (P) restricted to the degenerate rows. Although the tableau would not be explicitly available in an advanced implementation of the Revised Simplex Method, the data for (DS+) can be reconstructed efficiently. Moreover, different cases involve only changing of the right-hand side. Therefore, shadow prices can be computed by an appropriate sequence of (DS_+) and (DS_-). Fig. 4 shows the output from an experimental code for our example.

8. Discussion

It is shown that conventional LP software does not provide complete shadow prices in general. Although marginal values can indeed be found using repeated applications of existing codes, it would be desirable to automate such computations in commercial packages. This can be done without extensive modifications. In view of the fact that dual variables are routinely interpreted (often incorrectly as we have seen) as shadow prices by practitioners, it is important that only truly meaningful information is generated.

Author's Note

This article is based on unpublished research completed in 1987 at the University of Tennessee, Knoxville, with the assistance of Darren Smith. While the software used is outdated, the principles and results remain intact.

References

Akgül M. (1984). A note on shadow prices in linear programming. *J. Opl. Res. Soc.*, **35**, 425–431.

Aucamp, D.C., D.I. Steinberg (1982). The computation of shadow prices in linear programming. *J. Opl. Res. Soc.*, **33**, 557–565.

Best, M.J. (1982). A compact formulation of an elastoplastic analysis problem, *J. Optim. Theory Appl.*, **37**, 343–353.

Dantzig, G.B. (1963). *Linear Programming and Extensions*. Princeton Universtiy Press.

Dantzig, G.B. (1978). Are dual variables prices? If not, how to make them more so. *TR SOL 78–6*, Systems Optimization Laboratory, Stanford University, USA; also (1981). In *Mathematical Programming and its Economic Applications*, Franco Angeli Editore, pp. 135–148.

Evans J.R., N.R. Baker (1982). Degeneracy and the (mis) interpretation of sensitivity analysis in linear programming. *Decision Sciences*, **13**, 148–154.

Gal T. (1979) *Postoptimality Analysis, Parametric Programming and Related Topics*. McGraw-Hill, New York.

Gal T. (1986). Shadow prices and sensitivity analysis in linear programming under degeneracy: state-of-the-art survey. *OR Spektrum*, **8**, 59–71.

Gauvin J. (1980). Quelques précisions sur les prix marginaux en programmation linéaire. *INFOR*, **18**, 68–73.

Greenberg H.J. (1986). An analysis of degeneracy. *Naval Research Logistics Quarterly*, **33**, 635–655.

Ho J.K. (1987). *Linear and Dynamic Programming with Lotus 1–2–3*. MIS Press.

IBM, MPSX/370 (1979). Program Reference Manual.

Mangasarian O.L. (1979). Uniqueness of solution in linear programming. *Lin. Alg.& Appl.*, **25**, 151–162.

Murty K.G. (1983). *Linear Programming*. Wiley.

Peterson E.L. (1970). An economic interpretation of duality in linear programming. *J. Math. Anal. Appl.*, **30**, 172–196.

Pérold A.F. (1981). Exploiting degeneracy in the Simplex method. In G.B. Dantzig *et al.* (Eds.) *Large-Scale Linear Programming*. IIASA, Laxenburg.

Rockafellar T. (1970). *Convex Analysis*. Princeton University Press.

Rubin D.S., H.M. Wagner (1990). Shadow prices: tips and traps for managers and instructors. *Interfaces*, **20**, 150–157.

Schrage L. (1986). *Linear, Integer, and Quadratic Programming with LINDO*. Third Edition, Scientific Press.

Shapiro J.F. (1979). *Mathematical Programming: Structure and Algorithms*. Wiley.

Williams A.C. (1963). Marginal values in linear programming. *SIAM*, **11**, 82–94.

J.K. Ho is a professor of Information & Decision Sciences at the University of Illinois at Chicago. A 1970 graduate of Columbia University, he obtained his doctorate at Stanford University in 1974. His research interest is in information management and systems optimization for electronic commerce, with the latest book *Cyber Tigers: How Companies in Asia Can Prosper from E-commerce* (Prentice Hall/Pearson Education, Singapore: 2000; Chinese translation: SCMP Book Pub Ltd, Hong Kong: 2000).

Tikslių šešėlinių kainų skaičiavimas tiesiniame programvime

James K. HO

Yra žinoma, kad tiesiniame programavime dualių kintamųjų optimalios reikšmės gali būti. interpretuojamos kaip dešinės pusės koeficientų šešėlinės kainos (kraštinės reikšmės). Toks dualių kintamųjų ir šešėlinių kainų ekvivalentiškumas galimas tik esant neišsigimimo prielaidai. Esamos programinės sistemos dažniausia šios prielaidos netikrina. Straipsnyje apžvelgiamo ir apibendrinamo kylančios problemos ir demonstruojama, kaip tikslios šešėlinės kainos gali būti skaičiuojamos modifikuojant turimą programinę įrangą arba be modifikavimo.