6.1

Revised Primal Simplex method

Katta G. Murty, IOE 510, LP, U. Of Michigan, Ann Arbor

First put LP in *standard form.* This involves floowing steps.

- If a variable has only a lower bound restriction, or only an upper bound restriction, replace it by the corresponding nonnegative slack variable.
- If a variable has both a lower bound and an upper bound restriction, transform lower bound to zero, and list upper bound restriction as a constraint (for this version of algorithm only. In bounded variable simplex method both lower and upper bound restrictions are treated as restrictions, and not as constraints).
- Convert all inequality constraints as equations by introducing appropriate nonnegative slack for each.
- If there are any unrestricted variables, eliminate each of them one by one by performing a pivot step. Each of these reduces no. of variables by one, and no. of constraints by one. This

is equivalent to having them as *permanent basic variables* in the tableau.

- Write obj. in min form, and introduce it as bottom row of original tableau.

- Make all RHS constants in remaining constraints nonnegative.

Example: Max $z' = x_1 - x_2 + x_3 + x_5$

subject to $x_1 - x_2 - x_4 - x_5 \geq 2$

$$x_2 - x_3 + x_5 + x_6 \leq 11$$

$$x_1 + x_2 + x_3 - x_5 = 14$$

$$-x_1 + x_4 = 6$$

$$x_1 \geq 1 \text{ ; } x_2 \leq 1; \; x_3, x_4 \geq 0; \; x_5, x_6 \text{ unrestricted.}$$

Revised Primal Simplex Algorithm With Explicit Basis Inverse.

INPUT NEEDED: Problem in standard form, original tableau, and a primal feasible basic vector.

Original Tableau

x_1	$\dots$	x_j	$\dots$	x_n	$-z$	
a_{11}	$\dots$	a_{1j}	$\dots$	a_{1n}	0	b_1
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$
a_{m1}	$\dots$	a_{mj}	$\dots$	a_{mn}	0	b_m
c_1	$\dots$	c_j	$\dots$	c_n	1	α

Initial Setup: Let x_B be primal feasible basic vector and $B_{m\times m}$ be associated basis. Get augmented basic vector, listing $-z$ as basic var. in bottom (objective) row.

$$\text{Augmented Basis} \quad = \begin{pmatrix} B & 0 \\ c_B & 1 \end{pmatrix}$$

$$\text{Inverse Tableau} \quad = \begin{pmatrix} B & 0 \\ c_B & 1 \end{pmatrix}^{-1} = \begin{pmatrix} B^{-1} & 0 \\ -\pi & 1 \end{pmatrix}$$

$$\text{Basic values} = \begin{pmatrix} x_B \\ -z \end{pmatrix} = (\text{Inverse tableau}) \begin{pmatrix} b \\ \alpha \end{pmatrix} = \begin{pmatrix} \bar{b} \\ -z^0 \end{pmatrix}$$

where $\bar{b} \geq 0$ by primal feasibility. Then information at this stage stored in form of a tableau, also called inverse tableau.

Inverse tableau wrt x_B

Basic var.	Inverse tableau		Basic values
x_B	B^{-1}	0	$\bar{b}_1$ $\vdots$ $\bar{b}_i$ $\vdots$ $\bar{b}_m$
$-z$	$-\bar{\pi}$	1	$-z^0$

All nonbasics = 0

How to Update Inverse When a Column Changes?

$B_{m\times m}$, B^{-1} available. Suppose $B_{.j}$ replaced by column $d \in R^m$

leading to $\hat{B}$. How to obtain $\hat{B}^{-1}$ efficiently from B_{-1}? This needs only one pivot step.

Compute updated col. of d = representation of d in terms of $B = B^{-1}d = \bar{d} = (\bar{d}_1, \dots, \bar{d}_m)^T$, say.

If $\bar{d}_j = 0$, $\hat{B}^{-1}$ does not exist because $\hat{B}$ is singular.

If $\bar{d}_j \neq 0$, put B_{-1} in a tableau, and $\bar{d}$ on its RHS. Perform a GJ pivot step with $\bar{d}$ as pivot column and $\bar{d}_j$ as pivot element. This transforms B^{-1} into $\hat{B}^{-1}$.

Example:

$$B = \begin{pmatrix} 0 & -2 & 1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix}, B^{-1} = \begin{pmatrix} 1 & -1 & 2 \\ -1 & 1 & -1 \\ 1 & -2 & 2 \end{pmatrix}, j = 2, d = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$$

General Iteration:

- Compute rel. cost coeff. of nonbasic x_j

$$= \bar{c}_j = c_j - \pi A_{.j} = (-\pi, 1)\begin{pmatrix} A_{.j} \\ c_j \end{pmatrix}$$

= dot product of last row of inverse tableau with original col. of x_j.

- OPT. CRITERION: If all nonbasic $\bar{c}_j \geq 0$, present BFS opt., terminate.

- Otherwise, let $E = \{j : \bar{c}_j < 0\}$. Each x_j for $j \in E$ is *eligible to enter basic vector* x_B. Select one of them as *ENTERING VARIABLE*, say x_s.

- Compute updated col. of x_s =

$$\begin{pmatrix} \bar{A}_{.s} \\ \bar{c}_s \end{pmatrix} = B^{-1}\begin{pmatrix} A_{.s} \\ c_s \end{pmatrix}.$$

If $\bar{A}_{.s} \leq 0$, terminate, $z \longrightarrow -\infty$ in the problem.

If $\bar{A}_{.s} \not\leq 0$, perform pivot step by putting *pivot col.* = updated col. of x_s on right side of inverse tableau, selecting pivot row (dropping basic var.) by min ratio test, and performing a GJ pivot step. After piovot step, drop pivot col. and with new inverse tableau associated with new basic vector go to next iteration.

Examples:

Original tableau

x_1	x_2	x_3	x_4	x_5	x_6	$-z$	b
1	0	0	0	-1	1	0	2
0	1	0	0	1	1	0	1
0	0	1	0	2	1	0	5
0	0	0	1	0	1	0	0
0	0	1	1	-1	-8	1	0

$x_j \geq 0$ for all j, min z

Original tableau

x_1	x_2	x_3	x_4	x_5	x_6	$-z$	b
0	0	1	1	-1	-5	0	7
1	0	0	-1	-1	-3	0	9
0	1	0	-1	-1	0	0	1
-1	-1	-1	10	6	4	1	0

$x_j \geq 0$ for all j, min z

6.8

Revised Primal Simplex Method With Explicit Basis Inverse.

INPUT NEEDED: Problem in standard form, original tableau.

INITIAL SETUP: Look for unit basic vector. If found, let it be x_B. Starting with x_B as feasible basic vector, apply revised simplex algo. to solve problem (called Phase II).

If no unit basic vector, introduce nonnegative artificial variables with missing unit cols., and construct a unit basic vector with these artificial vars. in it. Let this basic vector be x_B. Define

$d_B =$ Basic phase I cost row vector (0 for original variables, 1 for artificial variables).

Augmented basic vector for Phase I is $(x_B, -z, -w)$, where

$w =$ Phase I objective function = sum of artificial variables, an *infeasibility measure* of the present solution.

Phase I Original Tableau

x_1	$\dots$	x_j	$\dots$	x_n	$-z$	$-w$	
a_{11}	$\dots$	a_{1m}	$\dots$	a_{1n}	0	0	b_1
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_{m1}	$\dots$	a_{mj}	$\dots$	a_{mn}	0	0	b_m
c_1	$\dots$	c_j	$\dots$	c_n	1	0	α
0	$\dots$	0	$\dots$	0	0	1	0

Only original problem variables (including slack variables) are displayed in Phase I original tableau. Artificials will never be considered as entering vars., so their data not needed in original tableau.

Phase I Inverse tableau wrt x_B

Basic var.	Inverse tableau			Basic values
x_B	B^{-1}	0	0	$\bar{b}_1$ $\vdots$ $\bar{b}_i$ $\vdots$ $\bar{b}_m$
$-z$	$-\bar{\pi}$	1	0	$-z^0$
$-w$	$-\sigma$	0	1	$-\bar{w}$

All nonbasics = 0

σ is Phase I basic dual solution. Apply primal simplex algo. to solve Phase I problem (minimize w). During Phase I entering var. choosen using Phase I rel. cost coeffs.

$\bar{d}_j$ = (boottom row of inverse tableau)(col. of x_j in Phase I original tableau)

PHASE I TERMINATION CRITERION: $\bar{d}_j \geq 0$ for all $j = 1$ to n.

At Phase I termination:

- if value of $w > 0$, original model infeasible, terminate. Go to *Infeasibility Analysis*.

- If value of $w = 0$, move from Phase I to Phase II to min original obj. func. z.

Transition From Phase I to Phase II:

- If all artificials left basic vector, delete last row ($-w$ row) from inverse tableau, and the last col. ($(m+2)$th) from inverse itself in final Phase I tableau. What remains is inverse tableau wrt present basic vector for original LP. Begin Phase II with it.

- If some artificials still in basic vector, their values must be 0 (otherwise $w \neq 0$).

  Fod any j with $\bar{d}_j > 0$, fix $x_j = 0$ and delete it from further consideration from original tableau.

  Delete last row from inverse tableau, and last row from inverse itself, and go to Phase II with resulting tableau. In this process all artificials in basic vector will remain = 0, or leave basic vector.

Examples

Original Tableau

x_1	x_2	x_3	x_4	x_5	$-z$	b
2	3	1	−1	0	0	10
1	2	−1	0	1	0	5
1	1	2	0	0	0	4
1	2	3	0	0	1	0

$x_j \geq 0$ for all j, minimize z

Original tableau

x_1	x_2	x_3	x_4	x_5	x_6	$-z$	b
1	−1	0	0	2	0	0	0
−2	1	0	0	−2	0	0	0
1	0	1	0	1	−1	0	3
0	2	1	1	2	1	0	4
−40	−10	0	0	−7	−14	1	0

$x_j \geq 0$ for all j, minimize z

To Find a Feasible Solution of a System of Linear Constraints

Gaussian elimination can find an unrestricted sol. of a system of linear eqs.

When system involves ineq. constraints, and/or bounds on variables, to find a feasible sol. we put the system in standard form (eqs. in nonnegative vars.) and apply Phase I of simplex method to find a feasible sol.

Infeasibility Analysis

Many ways to modify an infeasible model to a feasible one. In most applications, the preferred way is to alter the RHS constants. Most logical way is to change RHS constants, so that sum of absolute values of changes is minimum.

Consider model in standard form: $Ax = b, x \geq 0,$ where $b = (b_1, \ldots, b_m)^T$. Let $b' = (b'_1, \ldots, b'_m)$ where

$$b'_i = \begin{cases} b_i & \text{if } i\text{th basic var. in final phase I tableau is} \\ & \text{an original problem var. (or slack)} \\ b_i - v_i & \text{if } i\text{th basic var. in final phase I tableau is} \\ & \text{an artificial var. with value } v_i \end{cases}$$

Then changing b to b' would convert the model into a feasible one while minimizing the sum of absolute changes.

Examples: First problem considered under Simplex Method: $b = (10, 5, 4)^T$.

Basic var.	Inverse tableau					Basic values
t_1	1	-1	-1	0	0	1
x_2	0	1	-1	0	0	1
x_1	0	-1	2	0	0	3
$-z$	0	-1	0	1	0	-5
$-w$	-1	1	1	0	1	-1

b	basic	Inverse tableau							
3	x_6	1	1	0	-1	-1	0	0	6
0	x_{10}	0	-1	1	0	0	0	0	9
6	t_3	0	2	2	1	0	0	0	4
7	x_2	1	0	0	-1	1	0	0	5
10	t_2	-2	1	0	0	1	0	0	3
	$-z$	-1	-2	0	0	4	1	0	-100
	$-w$	-2	-1	1	-1	0	0	1	-7

How to Compute Marginal Values (MV) in a General LP Model ?

(P) original LP Model. (P') equivalent LP in standard form. So, obj. func. in (P') is in min form.

Assume (P') has a nondegenerate optimum BFS. Then (P') has a MV vector which is its opt. dual sol. Now how to find MV vector for (P)? Let x_B be opt. basic vector for (P').

CASE 1: No unrestricted variables in (P):

The MV associated with a "$\geq$" ["$\leq$"] inequality in (P) is the relative cost coeff. wrt x_B of the corresponding slack variable in (P') if obj. func. in (P) is in min [max] form, or its negative if obj. func. in (P) is in max [min] form.

The MV associated with an eq. constraint in (P) is same as its MV in (P') if obj. func. in (P) is in min form, or its negative if obj. func. in (P) is in max. form.

CASE 2: Unrestricted variables in (P), eliminated to get (P'):

Let Tableau A refer to tableau consisting of system of equality constraints just before the unrestricted variables were eliminated, with bottom row as obj. row in min. form.

When you augment opt. basic vector x_B for (P') with all the unrestricted vars. that were eliminated, you get a vector x_F, say, which will be a basic vector for Tableau A. Compute the dual basic sol. wrt x_F for Tableau A by usual formula. Values of dual vars. associated with equality constraints in (P) in this dual sol., are the MV [negatives of MV] corresponding to these constraints in (P) if obj. func. there is in min [max] form.

Revised Simplex Method Using Product Form OF Inverse

Pivot Matrices

Performing the G J Pivot step on a mtrix or tableau with a_{rs} as pivot element and following as pivot column

$$\begin{array}{c} \text{PC} \\ \hline a_{1s} \\ \vdots \\ \boxed{a_{rs}} \\ \vdots \\ a_{ms} \\ \hline \end{array}$$

is equivalent to multiplying matrix on left by following matrix of order m which differs from unit matrix only in its rth column.

$$P = \begin{pmatrix} 1 & \dots & 0 & -a_{1s}/a_{rs} & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 1 & -a_{r-1,s}/a_{rs} & 0 & \dots & 0 \\ 0 & \dots & 0 & 1/a_{rs} & 0 & \dots & 0 \\ 0 & \dots & 0 & -a_{r+1,s}/a_{rs} & 1 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & -a_{ms}/a_{rs} & 0 & \dots & 1 \end{pmatrix}$$

That's why P is called a *pivot matrix*. Its rth column (the only column in it different from a unit column) is called its *eta vector*.

Example

$$\begin{array}{rrrr|r} \hline 7 & 0 & -1 & 1 & 10 \\ 4 & -6 & \boxed{2} & 2 & 4 \\ 5 & -2 & 1 & 0 & 15 \\ \hline \end{array}$$

The Revised Simplex method always starts with unit basis corresponding to a basic vector which may contain some artificial variables. So initial inverse tableau, denote it by P_0, differs from unit matrix in last row only if we begin in Phase II; or in last two rows only, if we begin in Phase I. So, P_0 can be stored very efficiently, by storing these one or two rows.

In this implementation, inverse tableau not stored explicitly, but pivot matrices corresponding to all pivot steps carried so far are stored in the order in which they occured. Each pivot matrix differs from unit matrix in just one column (its *eta column*), it can be stored compactly by storing that col., and its position in the matrix.

Let $P_1, \ldots, P_r$ be the pivot matrices in that order at some stage.

Only use we make of inverse tableau is: its last row to compute nonbasic rel. costs, and then to compute pivot col. = updated col of entering var.

The inverse tableau itself is $P_r P_{r-1} \ldots P_1 P_0$. So,

Last row of inverse tableau $= (0, \ldots, 0, 1)P_r \ldots P_1 P_0$

Updated col. of entering var. $x_s = P_r \ldots P_1 P_0 \mathcal{A}_{.s}$

where $\mathcal{A}_{.s}$ is the original col. of entering var. x_s.

The first product above can be computed one pivot matrix at a time from left to right. Similarly the 2nd product above can be computed one pivot matrix at a time from right to left.

The sequence of pivot matrices $P_0, P_1, \ldots, P_r$ in order of occurrence is called the *string of pivot matrices at this stage*, and the computation outlined above is called *string computations*.

Reinversions

Roundoff errors accumulate in the inverse or string.

If x_B is present basic vector, with B, c_B as basis and basic cost vector, and $\bar{x}, \bar{\pi}$ as present primal and dual basic solutions; we monitor effect of roundoff errors by computing $||A\bar{x} - b||, \quad ||c_B - \bar{\pi}B||$. If these are smaller than specified tolerances, we continue. Otherwise, we reinvert the basis B and reconstruct the explicit inverse or the string, to replace the present quantities, and continue.