MATH 340: ETA MATRICES, THE ASSIGNMENT PROBLEM, AND OTHER APPLICATIONS

JOEL FRIEDMAN

Contents

The point of this article is to summarize what was covered in class in Math 340 after Chapters 1–5 of Vanderbei's text [Van14] (the simplex method). Although most of these topics are discussed somewhere in [Van14], the textbook often gives too many details or too vast a general setting to make it easy to understand the concepts; here we make the discussion brief—as was done in class—to make the important ideas as easy to understand as possible.

At this point this set of notes covers:

(1) eta matrices (in [Van14], Section 8.3) and the revised simplex method (in [Van14], Chapter 8);
(2) applications, including
   (a) the Assignment Problem (i.e., Weighted Bipartite Matching) (in Section 15.2 of [Van14], which relies on the more general discussion of Chapter 14);
   (b) matrix games, including our poker game where Alice has 2^{52} possible strategies;

Research supported in part by an NSERC grant.

(c) other applications covered in the exercises;
(d) remarks in the exercises about linear programs with unrestricted variables (i.e., that can take on negative and positive values): although in standard form we express each unrestricted variable as the difference of two non-negative variables, this does not mean that we are doing twice the work;
(e) similar remarks when we have constraints that are equalities instead of inequalities.

1. Eta Matrices, Tableaux, and the Revised Simplex Method

Matrix notation is covered in Section 6.1 of Vanderbei, and eta matrices in Section 8.3. We wish to make some remarks about this, and discuss "tableaux," which are not discussed in Vanderbei's text.

1.1. **Tableaux and Eta Matrices.**

Consider the LP that we use over and over in this class,

$$\text{maximize } \zeta = 4x_1 + 5x_2, \quad \text{subject to}$$
$$\begin{array}{rcl} x_1 + x_2 & \le & 5, \\ x_1 + 2x_2 & \le & 8, \\ 2x_1 + x_2 & \le & 8, \end{array}$$
$$\text{and} \qquad x_1, x_2 \ge 0.$$

with initial dictionary:

$$\begin{aligned} \zeta &= 4x_1 + 5x_2 \\ w_1 &= 5 - x_1 - x_2 \\ w_2 &= 8 - x_1 - 2x_2 \\ w_3 &= 8 - 2x_1 - x_2 \end{aligned}$$

In matrix notation we write

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

(we write x_3, x_4, x_5 for w_1, w_2, w_3) and write the dictionary as

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{bmatrix} \vec{x} = \vec{b} = \begin{bmatrix} 5 \\ 8 \\ 8 \end{bmatrix} \qquad (1)$$

Notice the identity matrix that appears as a submatrix of the big matrix to the left of $\vec{x}$; in class and in this article we sometimes refer to this big, 3×5 matrix as "big A."

How can we view the simplex method pivot "x_2 enters and x_4 (i.e., w_2) leaves"? (Note that for now we ignore the objective ζ.) With dictionaries, the equation

$$w_2 = 8 - x_1 - 2x_2$$

is convered to

$$x_2 = 4 - (1/2)x_1 - (1/2)w_2.$$

In the style of (1) we are converting the equation

$$x_1 + 2x_2 + w_2 = 8$$

to

$$(1/2)x_1 + x_2 + (1/2)w_2 = 4,$$

which is nothing but multiplying the second row of the matrix in (1) by 1/2. In augmented matrix notation we write the equations for $\vec{x}$:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & | & 5 \\ 1 & 2 & 0 & 1 & 0 & | & 8 \\ 2 & 1 & 0 & 0 & 1 & | & 8 \end{bmatrix}$$

in the equivalent system of equations

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & | & 5 \\ 1 & 2 & 0 & 1 & 0 & | & 8 \\ 2 & 1 & 0 & 0 & 1 & | & 8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & | & 5 \\ 1/2 & 1 & 0 & 1/2 & 0 & | & 4 \\ 2 & 1 & 0 & 0 & 1 & | & 8 \end{bmatrix}$$

Notice that the row operation "leave rows 1 and 3 alone, and multiply row 2 by 1/2" corresponds to the leftmost matrix above, a 3×3 matrix representing these operations; this multiplication yields an equivalent set of equations.

The remaining basic variables, $x_3 = w_1$ and $x_5 = w_3$ are then, in the dictionary point of view, expressed in terms of x_1 and $x_4 = w_2$; in terms of the augmented matrix this corresponds to clear out the top and bottom entries of the second column (corresponding to x_2). In other words the new dictionary is given by the set of row operations indicated by the 3×3 matrix

$$\begin{bmatrix} 1 & -1/2 & 0 \\ 0 & 1/2 & 0 \\ 0 & -1/2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & | & 5 \\ 1 & 2 & 0 & 1 & 0 & | & 8 \\ 2 & 1 & 0 & 0 & 1 & | & 8 \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 1 & -1/2 & 0 & | & 1 \\ 1/2 & 1 & 0 & 1/2 & 0 & | & 4 \\ 3/2 & 0 & 0 & -1/2 & 1 & | & 4 \end{bmatrix}$$

Notice that the new dictionary is expressed as the rightmost matrix above, and that columns 3, 2, and 5 (in that order) form an identity matrix. We could convert the above to a dictionary where x_3, x_2, x_5 are basic, and x_1, x_4 are non-basic. We abbreviate the above matrix multiplication as

$$E_1 D_1 = D_2,$$

where D_1 is the augmented matrix for the first dictionary, D_2 for the second, and E_1 is the "eta matrix"

$$E_1 = \begin{bmatrix} 1 & -1/2 & 0 \\ 0 & 1/2 & 0 \\ 0 & -1/2 & 1 \end{bmatrix}$$

Definition 1.1. By an *eta matrix* we mean a square matrix which has the same columns as the identity matrix on all but one of its columns, say its i-th column, and whose i,i entry is non-zero.

The following theorem is easy to see.

Theorem 1.2. *Any eta matrix, E, is invertible, and its inverse, E^{-1}, is an eta matrix with the same non-identity column as in E.*

In general, we see that if we pivot again, then the next dictionary, D_3, is given as

$$E_2 D_2 = D_3$$

where E_2 is another eta matrix. For example, if we examine the objective

$$\zeta = 4x_1 + 5x_2 = 4x_1 + 5(4 - (1/2)x_1 - (1/2)x_4) = 20 + (3/2)x_1 - (5/2)x_4,$$

we want x_1 to enter, and hence $x_3 = w_1$ to leave, we have "x_1 enters and x_3 leaves" corresponds to the row operations given by the eta matrix

$$E_2 = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}$$

which transforms the first column of D_2 into $[1\ 0\ 0]^{\mathrm{T}}$; hence we obtain the third dictionary matrix, D_3, as

$$D_3 = E_2 D_2 = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 & 1 & -1/2 & 0 & | & 1 \\ 1/2 & 1 & 0 & 1/2 & 0 & | & 4 \\ 3/2 & 0 & 0 & -1/2 & 1 & | & 4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 2 & -1 & 0 & | & 2 \\ 0 & 1 & -1 & 1 & 0 & | & 3 \\ 0 & 0 & -3 & 1 & 1 & | & 1 \end{bmatrix}$$

This is just our familiar final dictionary rows where we express $x_1, x_2, x_5 = w_3$ in terms of $x_3 = w_1$ and $x_4 = w_2$.

Notice that we can also write

$$D_3 = E_2 D_2 = E_2 E_1 D_1,$$

and so two pivots can be represented as a product of two appropriate eta matrices.

In order to encorporate the objective ζ into this augmented matrix notation, we may write the ζ line as $4x_1 + 5x_2$ in the bottom of any dictionary; to do this in the third dictionary we write

$$\begin{bmatrix} 1 & 0 & 2 & -1 & 0 & | & 2 \\ 0 & 1 & -1 & 1 & 0 & | & 3 \\ 0 & 0 & -3 & 1 & 1 & | & 1 \\ \hline 4 & 5 & 0 & 0 & 0 & | & 0 \end{bmatrix}$$

Of course, this doesn't look right because we want to write ζ in terms of the non-basic variables; this corresponds to clearing out the 4 and 5 and 0 in the basic variables by row operations: we subtract 4 times row 1 and 5 times row 2 from the ζ line to get

$$\left[\begin{array}{ccccc|c} 1 & 0 & 2 & -1 & 0 & 2 \\ 0 & 1 & -1 & 1 & 0 & 3 \\ 0 & 0 & -3 & 1 & 1 & 1 \\ \hline 0 & 0 & -3 & -1 & 0 & -23 \end{array}\right]$$

This array of numbers should look very (comfortingly) familiar; the 23 has a negative sign because this bottom line really says

$$-3x_3 - x_4 = -23 + \zeta;$$

can also write the bottom row with negative signs, but you must remember which is which. You can also view ζ as a variable. In class, I will use the above convention.

We may write all the dictionaries in these tables of numbers, which are generally called *tableaux* (the plural of the French *tableau*).

1.2. **Tableaux Example.** Let us summarize the simplex method on usual LP, adding the objective $\zeta = 4x_1 + 5x_2$ to each step. The initial dictionary

$$\begin{aligned} \zeta &= 4x_1 + 5x_2 \\ w_1 &= 5 - x_1 - x_2 \\ w_2 &= 8 - x_1 - 2x_2 \\ w_3 &= 8 - 2x_1 - x_2 \end{aligned}$$

has corresponding tableau

$$\left[\begin{array}{ccccc|c} 1 & 1 & 1 & 0 & 0 & 5 \\ 1 & 2 & 0 & 1 & 0 & 8 \\ 2 & 1 & 0 & 0 & 1 & 8 \\ \hline 4 & 5 & 0 & 0 & 0 & 0 \end{array}\right],$$

with the bottom row the ζ row. Then x_2 enters and $x_4 = w_2$ leaves means that we use row operations to convert column to $[0\ 1\ 0 \mid 0]^{\mathrm{T}}$, which is what the $x_4 = w_2$ looks like before the pivot. We get the second tableau

$$\left[\begin{array}{ccccc|c} 1/2 & 0 & 1 & -1/2 & 0 & 1 \\ 1/2 & 1 & 0 & 1/2 & 0 & 4 \\ 3/2 & 0 & 0 & -1/2 & 1 & 4 \\ \hline 3/2 & 0 & 0 & -5/2 & 0 & -20 \end{array}\right]$$

The second and final pivot, x_1 enters and $x_3 = w_1$ leaves gives the final tableau

$$\left[\begin{array}{ccccc|c} 1 & 0 & 2 & -1 & 0 & 2 \\ 0 & 1 & -1 & 1 & 0 & 3 \\ 0 & 0 & -3 & 1 & 1 & 1 \\ \hline 0 & 0 & -3 & -1 & 0 & -23 \end{array}\right]$$

1.3. **Matrix Notation and Eta Matrices.** Now let us see how this figures into our matrix notation. We write our dictionary constraints as:

$$A_B\vec{x}_B + A_N\vec{x}_N = \vec{b}$$

where A_B are the columns corresponding to the basic variables in the big matrix in (1), and A_N those corresponding to the non-basic variables. All our dictionaries can be derived from the above equation. So the first dictionary, derived from

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{bmatrix} \vec{x} = \vec{b} = \begin{bmatrix} 5 \\ 8 \\ 8 \end{bmatrix}$$

is expressed as

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \\ 8 \end{bmatrix}$$

It is more instructive to consider the second dictionary, where x_3, x_2, x_5 are basic and x_1, x_4 are non-basic. This is, therefore:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_3 \\ x_2 \\ x_5 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \\ 8 \end{bmatrix}$$

We therefore get:

$$\begin{bmatrix} x_3 \\ x_2 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}^{-1} \left(\begin{bmatrix} 5 \\ 8 \\ 8 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_4 \end{bmatrix} \right)$$

Curiously (or not so curiously) we note that

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -1/2 & 0 \\ 0 & 1/2 & 0 \\ 0 & -1/2 & 1 \end{bmatrix}$$

which is just the matrix we called E_1. Similarly, the third dictionary, with x_1, x_2, x_5 in the basis can be determined by:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \\ 8 \end{bmatrix};$$

not surprisingly, we have

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 2 & 1 & 1 \end{bmatrix}^{-1}$$

is the matrix given by E_2E_1 above. In other words we have

$$A_{B_1}^{-1} = E_1 A_{B_0}^{-1},$$

and

$$A_{B_2}^{-1} = E_2 A_{B_1}^{-1},$$

where B_0, B_1, B_2 are the basic variables at step $1, 2, 3$ (respectively) of the simplex method, and the E_i were the eta matrices describing the row operations above.

It is clear that this discussion generalizes to any form of the simplex method. Let us record these observations.

Theorem 1.3. *Let B_i be the set of basic variables on the i-th iteration of this simplex method, and let A_{B_i} be the part of the matrix "big A" that corresponds to the basic variables in the i-th stage of the simplex method. Then for any positive integer i, we have*

$$A_{B_i}^{-1} = E_i A_{B_{i-1}}^{-1}$$

where E_i is an eta matrix representing the row operations used to convert the i-th tableau to the $(i+1)$-st tableau.

Corollary 1.4. *If A_{B_i} is as in the above theorem, then A_{B_i} is invertible for all i. In particular,*

(1) there is no way to linearly combine the rows of A_{B_i} non-trivially and get a row of all zeros; and
(2) the same, with "rows" replaced with "columns."

The latter statements in this corollary follow from the fact that if the rows or columns of a square matrix are linearly dependent, then the determinant of the matrix must be zero, and hence this matrix cannot have an inverse.

The above corollary is crucial to our discussion of weighted bipartite matching.

1.4. **The Revised Simplex Method.** Another application of the eta matrices is to the revised simplex method. The idea is as follows: the dictionaries will look like:

$$A_B \vec{x}_B + A_N \vec{x}_N = \vec{b},$$

$$\zeta = \vec{c} \cdot \vec{x} = \vec{c}_B \cdot \vec{x}_B + \vec{c}_N \cdot \vec{x}_N = \vec{c}_B^{\mathrm{T}} \vec{x}_B + \vec{c}_N^{\mathrm{T}} \vec{x}_N$$

(the transpose convention for dot products will be useful here). We can solve for $\vec{x}_B$ to get

$$\vec{x}_B = A_B^{-1}(\vec{b} - A_N \vec{x}_N),$$

$$\begin{aligned} \zeta &= \vec{c}_B^{\mathrm{T}} \vec{x}_B + \vec{c}_N^{\mathrm{T}} \vec{x}_N \\ &= \vec{c}_B^{\mathrm{T}} A_B^{-1}(\vec{b} - A_N \vec{x}_N) + \vec{c}_N^{\mathrm{T}} \vec{x}_N \\ &= \vec{c}_B^{\mathrm{T}} A_B^{-1} \vec{b} + (\vec{c}_N^{\mathrm{T}} - \vec{c}_B^{\mathrm{T}} A_B^{-1} A_N) \vec{x}_N \end{aligned}$$

Let us pause to summarize the above calculations.

Theorem 1.5. *If B is any set of basis variables in the simplex method, then A_B is invertible, and the dictionary associated to B is given by*

$$\begin{array}{ccccc} \vec{x}_B & = & A_B^{-1}\vec{b} & - & A_B^{-1}A_N & \vec{x}_N, \\ \zeta & = & \vec{c}_B^{\mathrm{T}}A_B^{-1}\vec{b} & + & (\vec{c}_N^{\mathrm{T}} - \vec{c}_B^{\mathrm{T}}A_B^{-1}A_N) & \vec{x}_N. \end{array}$$

In practice we don't want to write down every entry of this dictionary, rather we only write down those we need for each pivot. As these formulas suggest, the most laborious task is to compute A_B^{-1} times various quantities.

Remark 1.6. Generally we only need to compute the ζ row to determine which variable enters, and then compute the dictionary entries of this entering variable (and the dictionary constants) to see which variable leaves. By the above formulas we will want to compute the ζ row coefficients:

$$\vec{c}_B^{\mathrm{T}}A_B^{-1}A_N.$$

We note that we compute this as

$$(\vec{c}_B^{\mathrm{T}}A_B^{-1})A_N, \quad \text{not} \quad \vec{c}_B^{\mathrm{T}}(A_B^{-1}A_N).$$

The latter way is much more expensive.

For the first pivots we might just use:

$$A_{B_1}^{-1} = I, \quad A_{B_2}^{-1} = E_1, \quad A_{B_3}^{-1} = E_2E_1, \quad \text{etc.}$$

So that we compute

$$\vec{c}_B^{\mathrm{T}}A_{B_3}^{-1}$$

as

$$\vec{c}_B^{\mathrm{T}}E_2E_1.$$

After a number of iterations, two problems arise:

(1) it is no longer a savings to use

$$A_{B_i}^{-1} = E_{i-1}E_{i-1}\dots E_1$$

for very large i, and

(2) the formula

$$A_{B_i}^{-1} = E_{i-1}E_{i-1}\dots E_1$$

tends to introduce a lot of roundoff error.

It turns out that both of these issues can be important, depending on the problem.

There are a number of subtleties here. For one, we never actually write down the whole matrix, $A_{B_i}^{-1}$, rather we factor A_{B_i} as LU, with L lower triangular and U upper triangular, and use this "LU-decomposition" to compute $A_{B_i}^{-1}$ when applied to vectors. So every so often we compute $A_{B_i}^{-1}$ from scratch, and then use

$$A_{B_{i+1}}^{-1} = E_iA_{B_i}^{-1}, \quad A_{B_{i+2}}^{-1} = E_{i+1}E_iA_{B_i}^{-1},$$

and so on, until the repeated use of eta matrices is problematic; then we refactor the $A_{B_{i'}}$ (for some $i' > i$), and the use

$$A^{-1}_{B_{i'+1}} = E_{i'} A^{-1}_{B_{i'}}, \quad A^{-1}_{B_{i'+2}} = E_{i'+1} E_{i'} A^{-1}_{B_{i'}},$$

again for a while.

Assuming that we refactor the A_{B_i} every s steps, and that refactorizing an A_{B_i} takes m^3 time, and that applying A_{B_i} takes time m^2, the average cost over s steps is of order $m^3 s^{-1}$ (refactorizing every s steps), and ms (the last $s/2$ steps involve multiplying out at least $s/2$ matrices), we see that $m^3 s^{-1} + ms$ is optimal for $s = m$ (where $m^3 s^{-1} + ms$ becomes order m^2). Furthermore, in Vanderbei's text it is claimed that in certain sparse problems the refactorizations require closer to m^2 steps, in which case s is optimal when of order $\sqrt{m}$, so that $m^2 s^{-1} + ms$ is over order $m^{3/2}$. Vanderbei's text does not perform any analysis to indicate when roundoff errors become a problem.

If we have a linear program in standard form with constraints $A\vec{x} \le \vec{b}$ where A is an $m \times n$ matrix, then we have n decision variables and m slack variables, and hence all the A_{B_i} matrices are $m \times m$. If m is large, then computing $A^{-1}_{B_i}$ is expensive, but each eta matrix, E_i, has only one column (of m numbers) that we need to remember, and hence multiplying by E_i requires only order m FLOPS (Floating Point OPerationS).

There are a number of details to get this method to work quickly, and a number of computational issues involved (see the rest of Chapter 8 of Vanderbei).

2. Applications

2.1. **Weighted Bipartite Matching.**

The Assignment Problem, or Weighted Bipartite Matching problem, discussed in class and in Section 15.2 of Vanderbei is the problem:

$$\text{maximize} \sum_{i,j=1}^{n} c_{ij} x_{ij}$$

subject to

$$\forall i, \quad x_{i1} + x_{i2} + \cdots + x_{i,n} = 1,$$

and

$$\forall j, \quad x_{1j} + x_{2j} + \cdots + x_{n,j} = 1,$$

and all $x_{ij} \ge 0$.

To simplify the linear program, we argue that we can add the same constant to all of the c_{ij} without altering the optimal choice of x_{ij}. Hence we may assume that $c_{ij} > 0$ for all i and j. In class we argued that we could weaken the constraints to

$$\forall i, \quad x_{i1} + x_{i2} + \cdots + x_{i,n} \le 1,$$

and

$$\forall j, \quad x_{1j} + x_{2j} + \cdots + x_{n,j} \le 1,$$

and all $x_{ij} \ge 0$.

We therefore get a dictionary with

$$\zeta = \sum_{i,j} c_{ij} x_{ij},$$

and slack variables

$$wrow_i = 1 - x_{i1} - x_{i2} - \cdots x_{in} \quad \text{for } i = 1, \ldots, n$$

and

$$wcol_j = 1 - x_{1j} - x_{2j} - \cdots x_{nj} \quad \text{for } j = 1, \ldots, n.$$

Since this problem is feasible and bounded (by n times the maximum of the c_{ij}), there is a final dictionary giving an optimal solution.

In class, we argued that each slack variable must be zero in an optimal solution (or else we could increase at least one of the x_{ij} and get a higher objective value).

The fact that

$$\sum_{i=1}^{n} wrow_i = \sum_{j=1}^{n} wcol_j$$

shows that we can combine the rows of the initial dictionary (or tableau) to get a tableau row that looks like

$$[0\ 0\ \cdots\ 0\ 1\ 1\ \cdots\ 1\ -1\ -1\ \cdots\ -1]$$

where the 1's correspond to the $wrow_i$ variables, and the -1's correspond to the $wcol_j$ variables. It follows that any simplex method basis variables $\vec{x}_B$ must contain at least one the the $wrow_i$ or $wcol_j$ variables, for otherwise the A_B columns can be combined non-trivially to get a row of all 0's in A_B, and hence A_B would not be invertible. This means that there are at most $2n-1$ of the x_{ij}'s in any basis of the simplex method, and hence any basis has at most one x_{ij} in one of its rows, and similarly in one of its columns. In the final dictionary it follows that in such a row and column there corresponds an x_{ij} that is the only non-zero variable in that row and column, and hence $x_{ij} = 1$.

From here one can argue that the simplex method (perhaps applied repeatedly to smaller problems) will generate a solution for which each x_{ij} is either one or zero.

However, there is a further remarkable property of the final dictionary of the simplex method running on a such a problem. It is mentioned in Chapter 14 of Vanderbei (in the general context of the "Network Simplex Method"), but not proven there.

Theorem 2.1. *Consider* any *dictionary of the linear program above for the $n \times n$ Weighted Bipartite Matching (i.e., the Assignment Problem) with the above notation. Consider the bipartite graph, G, with n vertices on the left* *and n on the right, and whose edges are the x_{ij} appearing in the basis. Then this graph has no cycles, i.e., this graph is a forest, i.e., this graph is a union of trees.*

Like our observation that *any* dictionary must have at least one $wrow_i$ or $wcol_j$ being a basic variable, we emphasize that this theorem is valid for *any* dictionary. Using this theorem it is easy to see that any final dictionary for the above simplex method has all the x_{ij}'s being one or zero. This reason, and the graph theory involved (cycles, trees, forests, etc.) will be discussed in class.

Proof. Each x_{ij} column appears in exactly two places: the $wrow_i$ and $wcol_j$ slack variable location, with a -1 coefficient in both places. A cycle in the above bipartite graph, say of length $2k$, is equivalent to having

$$x_{i_1,j_1}, x_{i_2,j_1}, x_{i_2,j_2}, x_{i_3,j_2}, \ldots x_{i_k,j_k}, x_{i_k,j_1}$$

as basic variables. But then the alternating sum of the columns of "big A" corresponding to the above variables

$$A_{x_{i_1,j_1}} - A_{x_{i_2,j_1}} + A_{x_{i_2,j_2}} - A_{x_{i_3,j_2}} \ldots + Ax_{i_k,j_k} - A_{x_{i_k,j_1}}$$

is a column of zeros. But this would show that A_B is not invertible. Hence the bipartite graph in the theorem contains no cycles.

It is a standard fact from graph theory that if G is a graph no cycles, then between any two vertices there is at most one path, and hence each connected component of G is a tree, i.e., G is a forest. □

Given the above theorem, consider a final dictionary and the above graph, G, for such a dictionary. Let v be any leaf (see the paragraph below) of the above graph, and say that v corresponds left vertex (which in class we called a "person") number i. Then it follows that there is a unique j such that x_{ij} is in the basis; hence if $j' \neq j$, $x_{ij'}$ is non-basic, and hence $x_{ij'} = 0$ in the corresponding BFS (Basic Feasible Solution). Hence $x_{ij} = 1$ (since the $wrow_i$ variable must be zero in the BFS for the optimal dictionary). It then follows that $x_{i'j} = 0$ for any $i' \neq i$ (such an $x_{i'j}$ may be basic or non-basic; so by removing left vertex (person) i and right vertex (which we called a "task" in class) j, we discard the edge x_{ij} of the graph, giving us a forest G' with one edge fewer, and we are still left with a forest. Now we can repeat this argument on G', finding an $x_{i'j'} = 1$, and discarding this edge to get G''. Eventually we discard all edges for G, giving us that $x_{ij} = 1$ for any edge of G. Hence in the final dictionary, all x_{ij} that are basic equal 1.

As far as the graph theory that we need, we made a few claims above that should seem reasonable. We really only need the following result: if $G = (V, E)$ is a graph, and the degree of a vertex, v (i.e., the number of edges incident upon v), is denoted $d(v)$, then we have

$$|E| - |V| = \sum_{v \in V} \bigl(2 - d(v)\bigr)/2.$$

For a tree (which is a connected graph without cycles) we have $|E| - |V| = -1$, and so at least one vertex of G must have $d(v) \leq 1$; if G is not an isolated vertex, then $d(v)$ is never zero, and hence there exist at least two vertices with $d(v) = 1$. A vertex with degree one is called a *leaf*.

2.2. **General Remarks About Game Theory.** We made a number of observations regarding game theory, some before the coverage of eta matrices.

First, the linear program giving the optimal strategy for Alice in "Alice announces a mixed strategy" is the one with Alice replaced by Betty.

Second, complementary slackness in linear programming makes the following simple observation about game theory: assume that $\vec{p}$ is Alice's optimal mixed strategy in a matrix game with matrix A, and that

$$\vec{p}A = [3\ 4\ 5\ 6\ 3\ 4];$$

hence the value of the mixed game is 3 (assuming that $\vec{p}$ is optimal). Then Betty, who has 6 strategies, will only play column 1 and 5 in an optimal strategy for her, for otherwise Betty will be unable to achieve the value 3. By writing down the linear programs for Alice's and Betty's optimal mixed strategies, we can chase through the complementary slackness picture to verify that this remark is just "half" of complementary slackness.

Third, imagine that we have a 100×4 matrix game with all positive entries, so that game "Alice announces a mixed strategy" amounts to the linear program maximize v subject to

$$[p_1\ \dots\ p_{100}]A \geq [v\ v\ v\ v], \quad p_1 + \cdots + p_{100} \leq 1$$

with the p_i and v being non-negative. Then the dictionaries for the simplex method on this linear program have 5 basic variables; in the final dictionary v must be basic, since v is positive in the optimal solution (v is as least as large as the smallest entry of A); this leaves at most 4 of the p_i that can be basic. It follows that there is an optimal strategy for Alice using at most 4 of the rows. A similar remark can be made for any shape of matrix.

2.3. **The Big Poker Question.** Consider the "big poker question" of Section 8 of the handout on game theory: (1) Alice and Betty each ante one penny; (2) Alice is dealt one of 52 cards; she looks at the card, but Betty cannot see the card; she either bets one penny or folds; (3) Betty either calls with one penny or folds. If Alice bets and Betty calls, then Betty draws a card and compares it to Alices card; the higher ranked card wins (in some order, say poker order).

Since Betty cannot see the card, Betty has only two pure strategies: call or fold. However, in a pure strategy Alice can elect to bet or fold; since she has to make a choice of these two options for 52 different cards, Alice has 2^{52} pure strategies.

In class we tried to reduce the number of possible strategies; we also wanted to see if our reductions would teach us some general principles.

The first observation is that if Alice bets on some set of cards, $\mathcal{B} \subset \{1, \ldots, 52\}$, then if we fix the size of $\mathcal{B}$ to $b = |\mathcal{B}|$, then for b fixed Alice does best to bet the top b cards. This shows us (something that we already suspected in September), that she only has 53 possible strategies, namely

$$\text{for } b = 0, 1, \ldots, 52, \text{ Alice bets } \{52, 51, \ldots, 52 - b + 1\}.$$

We also calculated that if Alice chooses to bet b cards, then her corresponding row in the matrix game looks like

$$\vec{f}(b) = \Bigl(f_1(b), f_2(b)\Bigr)$$

where $f_2(b)$ was a linear function of b (if Betty always folds when Alice bets, then Alice wins b times out of 52, and Betty wins $52 - b$ times out of 52), and where

$$f_1(b) = \frac{-4}{52 \cdot 51} b^2 + c_1 b + c_2$$

where c_1, c_2 were constants; in other words $f_1(b)$ is a quadratic function with a negative b^2 coefficient.

Then we made the following observations: in a matrix game, let Alice's b-th row be given as $\vec{f}(b)$ (the vector $\vec{f}$ has an many entries as there are columns in the matrix for the matrix game). Then

(1) if $\vec{f}(b)$ is a strictly convex function of b (i.e., $\vec{f}''$ is always non-negative in all its components, assuming that $\vec{f}$ is twice differentiable), and if there are m rows in the matrix, then there is an optimal strategy for Alice involving only rows 1 and m; and
(2) if $\vec{f}(b)$ is a strictly concave function of b (i.e., $\vec{f}''$ is always non-positive, assuming that $\vec{f}$ is twice differentiable), then there is a row r such that there is an optimal strategy for Alice involving only rows r and $r + 1$.

In general, $\vec{f}(b)$ is not necessarily convex or concave, and our observations are only valid assuming such properties.

3. Learning Goals and Sample Exam Problems

It seems best to make general learning goals concrete by connecting them with sample exam problems.

Learning Goals:

(1) You should be able to do the simplex method via tableaux, and write out the associated eta matrices. See Exercise 3.1, 3.2.
(2) You should know why eta matrices are important, and some of their properties. See Exercise **??**, **??**.

Sample Exam Problems

Exercise 3.1. Find the tableaux and eta matrices for our standard problem, given in (1), but where the first pivot has x_1 entering (instead of x_2). There will only be one choice of entering variable and leaving variable, and we should finish after three pivots. Do this in the style of Subsection 1.2, where the ζ line appears in the last row of the tableaux.

Exercise 3.2. Same as Exercise 3.1 (solve the linear program using tableaux, and write out the eta matrices) for the linear program:

$$\text{maximize } \zeta = x_1 + 7x_2, \quad \text{subject to}$$

$$\begin{array}{rcl} x_1 + x_2 & \leq & 3, \\ x_1 & \leq & 2, \\ x_2 & \leq & 2, \end{array}$$

$$\text{and} \qquad x_1, x_2 \geq 0.$$

Exercise 3.3. Consider the matrix game

$$A = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}.$$

Write down Alice's optimal mixed strategy for this game as a linear program, and use tableaux to find Alice's best strategy. [Hint: by looking at A, we see that Alice will play only row 2, since row 2 dominates row 1; so your first pivot should be v enters (there will be a tie for leaving variables in this degenerate pivot), and your second pivot shoul be p_2 enters, to save time.]

Exercise 3.4. Say that you are solving a linear program with 100 variables and 100 constraints. Explain why

(1) it may be advantageous to use eta matrices in the first 5 pivots:

$$A_{B_1}^{-1} = I, \quad A_{B_2}^{-1} = E_1, \quad \cdots, A_{B_5}^{-1} = E_4E_3E_2E_1,$$

and

(2) it may not be advantageous to use eta matrices for many pivots, such as

$$A_{B_{500}}^{-1} = E_{499}E_{498}\dots E_2E_1.$$

Explain why it may be advantegeous to recompute $A_{B_i}^{-1}$ every so often, and use eta matrices in between recomputations. [These recomputations are typically done by the LU-decomposition, not by writing down $A_{B_i}^{-1}$, but this is not important to us.]

Exercise 3.5. Consider the problem of 3-dimensional matching: intuitively, you have n people, n tasks, and n labs; to each assignment of one person to one task and one room, you are given a utility. You wish to maximize the total utility in an assignment, meaning that to each person you assign one task and one lab, so that all tasks are performed (each by one person) and all labs are used (each by one person). More concretely, you have a

3-dimensional array of real numbers c_{ijk}, and you want to choose x_{ijk} to be 0 or 1 such that

$$\sum_{i,j,k} c_{ijk} x_{ijk}$$

is maximized subject to

$$\text{for all } i, \quad \sum_{j,k} x_{ijk} = 1$$
$$\text{for all } j, \quad \sum_{i,k} x_{ijk} = 1$$
$$\text{for all } k, \quad \sum_{i,j} x_{ijk} = 1$$

Now consider the *relaxation* of the above problem, where each x_{ijk} can be a real number between 0 and 1.

(1) Argue that to find the optimal x_{ijk} you can assume that each c_{ijk} is positive.
(2) Argue that if all the c_{ijk} are positive, then you can replace the equalities above by inequalities.
(3) Argue that the resulting linear program is feasible and bounded.
(4) Argue that if we run the simplex method, then we can find an optimal solution such that at most $3n$ of the x_{ijk} are nonzero.

Remark: solving the original problem in which the x_{ijk} must all be 0 or 1 is an example of a problem that is "NP-complete." Hence if you can find a quick algorithm to determine the optimal solution subject to all x_{ijk} are either 0 or 1, you will be instantly famous and awarded a $1,000,000 (USD) prize.

Exercise 3.6. Consider the linear program to maximize $4x_1 + 5x_2$ subject to

$$\begin{aligned} x_1 + x_2 &\le 10 \\ 2x_1 + 2x_2 &\le 21 \\ 3x_1 + 3x_2 &\le 29 \\ x_1, x_2 &\ge 0 \end{aligned}$$

Argue that the simplex method will never yield a dictionary in which both x_1 and x_2 are basic; do this in two ways:

(1) by considering the columns of A_B, the basic part of "big A" corresponding to any dictionary; and
(2) by describing all possible ways that the simplex method can be run (e.g., the first pivot is either x_1 enters or x_2 enters).

Is the same true if the constants $10, 21, 29$ in the above linear program are replaced with any other constants?

Exercise 3.7. Consider the line $y = a + bx$ which is the best "max approximation" regression line to the data $(0, 4)$, $(1, 6)$, $(2, 7)$, $(3, 10)$. In other words, consider the optimization problem:

$$\min d \quad \text{subject to}$$
$$|4 - a| \le d,$$
$$|6 - a - b| \le d,$$
$$|7 - a - 2b| \le d,$$
$$|10 - a - 3b| \le d.$$

(1) Assume you can argue that d, a, b must all be strictly positive. Then write the above optimization problem as a linear program in standard form. [Hint: the inequality $|10 - a - 3b| \le d$ is the same as saying
$$10 - a - 3b \le d \quad \text{and} \quad 10 - a - 3b \ge -d.]$$
(2) Argue that the above linear program is feasible and bounded.
(3) Show that in any optimal solution (again assuming d, a, b are strictly), there must be three of the inequalities of your linear program that must be satisfied with equality. The three inequalities where the equalities are satisfied are known as "support points" (each such inequality/equality corresponds to one of the data points).
(4) Assume you are now looking for the best max-approximation of $y = a + bx$ to 10 data points (instead of the four data points above) that don't all exactly lie on a line $y = a + bx$), and say that you can argue that the best fit occurs with an a and b that are both positive. Can you still assert the existence of three "support points"?

Exercise 3.8. Consider the best "L^1 approximation" regression line $y = a + bx$ to the five data points $(0, 4)$, $(1, 6)$, $(2, 8)$, $(3, 11)$, $(4, 13)$. In other words, consider the optimization problem:

$$\min |4 - a| + |6 - a - b| + |8 - a - 2b| + |11 - a - 3b| + |13 - a - 4b|$$

over all real a, b.

(1) Assume you can argue that for the optimal a, b, both their values are positive. Write the above optimization problem as a linear program in standard form. [Hint: see Vanderbei, Section 12.4.]
(2) Ask something else?

Exercise 3.9. Consider any linear program in standard form such that the sum of the first two slack variables is equal to the sum of the third and fourth slack variables. Argue that the slack variables cannot all be non-basic in any dictionary of the simplex method. [Hint: Consider the first four rows of A_B, the basic part of "big A" corresponding to a basis B.]

Exercise 3.10. Consider the LP for the matrix game:

$$A = \begin{bmatrix} 2 & -5 \\ -4 & 3 \end{bmatrix},$$

namely

$$\begin{aligned} \max v \quad & \text{subject to} \\ 2p_1 - 4p_2 &\geq v, \\ -5p_1 + 3p_2 &\geq v, \\ p_1 + p_2 &= 1 \\ p_1, p_2 &\geq 0 \end{aligned}$$

Previously we added constants to each enty of A to make simplifying assumptions on the linear program. Here we ask if it is so bad just to leaves things as they are.

(1) Write the above linear program in standard form, where v is replaced by $v_1 - v_2$ where $v_1, v_2 \geq 0$ (since we don't know if v is positive or negative), and where $p_1 + p_2 = 1$ is replaced by the two inequalities:

$$p_1 + p_2 \leq 1 \quad \text{and} \quad -p_1 - p_2 \leq -1.$$

(2) Argue that this linear program is feasible and bounded, and therefore the simplex method will find the optimal value.
(3) Write down the "big A" matrix expressing the linear equations satisfied by the decision variables and the slack variables.
(4) What is the "big A" column corresponding to v_1, and the column corresponding to v_2? Argue that v_1 and v_2 can never both be basic variables in any dictionary given by the simplex method.
(5) Say that v_1 is a basic variable in some dictionary, and hence v_2 is non-basic. What will the v_1 row look like in the dictionary?
(6) Consider the two slack variables corresponding to the inequalities

$$p_1 + p_2 \leq 1 \quad \text{and} \quad -p_1 - p_2 \leq -1.$$

What is the sum of these two slack variables? Argue that at least one of these two slack must be basic in any dictionary that can arise in the simplex method.
(7) Generalize this discussion to any linear program where:
 (a) some variables are not known to be positive (or negative), and are written as the difference of two variables; and
 (b) some constraints are linear equalities, and hence are written as two inequalities.

Exercise 3.11. Consider a weighted bipartite matching problem between two people and two tasks, where the utility matrix c_{ij} has all positive entries.

(1) Write down a linear program for this problem and argue directly that at optimality each slack variable must equal zero.

(2) Argue directly that all four slack variables cannot be nonbasic in any dictionary of the simplex method. [Hint: Consider Exercise 3.9.]
(3) Argue that, on the basis of the previous part, that all four decision variables cannot be basic in any dictionary.
(4) Argue that, on the basis of the previous parts, the final dictionary will have two of the x_{ij} equal to one, and the other two equal to zero.

Exercise 3.12. Consider a weighted bipartite matching problem between two people and two tasks, where the utility matrix c_{ij} is given by $c_{11} = 8$, $c_{22} = 9$, $c_{12} = 5$, and $c_{21} = 3$. Since matchings between two people and two tasks can only be done one of two ways, it is clear that $x_{11} = x_{22} = 1$ and $x_{12} = x_{21} = 0$ is optimal; $x_{12} = x_{21} = 1$ and $x_{11} = x_{22} = 0$ is not optimal.
(1) Write down the complementary slackness conditions for the proposed optimal solution $x_{12} = x_{21} = 1$ and $x_{11} = x_{22} = 0$ (which we know is not optimal). Show that the four dual decision variables need only satisfy two equations, and hence there is no unique dual solution.
(2) Show that any solution for the four dual decision variables leads to an infeasible dual solution (i.e., at least one of the dual slack variables must be negative).

Exercise 3.13. Consider a weighted bipartite matching problem between two people and two tasks, where the utility matrix c_{ij} is given by $c_{11} = 8$, $c_{22} = 9$, $c_{12} = 5$, and $c_{21} = 3$. Since matchings between two people and two tasks can only be done one of two ways, it is clear that $x_{11} = x_{22} = 1$ and $x_{12} = x_{21} = 0$ is optimal; $x_{12} = x_{21} = 1$ and $x_{11} = x_{22} = 0$ is not optimal.
(1) Write down the complementary slackness conditions for the proposed optimal solution $x_{11} = x_{22} = 1$ and $x_{12} = x_{21} = 0$ (which we know is optimal). Show that the four dual decision variables need only satisfy two equations, and hence there is no unique dual solution.
(2) Find a dual solution (even though it is not unique) which satisfies complementary slackness and is dual feasible (i.e., all the y_i's and z_i's are non-negative, and they satisfy complementary slackness for the above x_i's and w_i's).

Exercise 3.14. Consider the matrix game:

$$A = \begin{bmatrix} 1 & 36 \\ 4 & 25 \\ 9 & 16 \\ 16 & 9 \\ 25 & 4 \\ 36 & 1 \end{bmatrix}$$

(1) Show that for any $i = 2, 3, 4, 5$ we have that playing 50% of row $i-1$ and 50% of row $i+1$ is better than playing row i.
(2) Show that if $\vec{f}(b)$ denotes the b-th row of this game, then $\vec{f}(b)$ is a convex (i.e., concave up) function of b.

(3) What two rows will Alice play in "Alice plays a mixed strategy," based on your answers to the previous parts?
(4) Find Betty's best mixed strategy.

Exercise 3.15. Consider the matrix game:

$$A = \begin{bmatrix} -1 & -36 \\ -4 & -25 \\ -9 & -16 \\ -16 & -9 \\ -25 & -4 \\ -36 & -1 \end{bmatrix}$$

(1) Show that if $\vec{f}(b)$ denotes the b-th row of this game, then $\vec{f}(b)$ is a concave down function of b.
(2) What two rows will Alice play in "Alice plays a mixed strategy," based on your answer to the previous part?
(3) Find Betty's best mixed strategy.

Exercise 3.16. Explain each of the following principles of the revised simplex method (they are all discussed in this article):

(1) We do not need to compute the whole dictionary: assume we want to compute the entire ζ row to choose which variable enters, but then compute constants of the dictionary and only the column of an entering variable, which computations are involved? [Hint: look at Theorem 1.5.]
(2) The order of multiplcation matters: Theorem 1.5 has a term $\vec{c}_B^{\mathrm{T}} A_B^{-1} A_N$. In which order do we multiply these terms?
(3) How are eta matrices used in the revised simplex method? Specifically, consider the cost in terms of FLOPS (FLOating Point Operations): if to refactorize A_B^{-1} requires m^α FLOPS (where m is the number of basic variables), where α ranges between 1 and 3 depending on the problem, and each multiplication by an eta matrix requires m FLOPS, argue that to refactorize every s steps would involve an average of order $m^\alpha s^{-1}$ FLOPS (for the refactorization) and order ms FLOPS for the eta matrix multiplications. Then argue that the optimal value of s is $m^{(\alpha-1)/2}$.

Solutions to some problems appearing two pages from now

Solutions to some problems appear starting on the next page

Here we give brief solutions to some problems in the article on Eta Matrices and Applications. This term problems 1,2,4–7,9 were assigned; here we give brief solutions to 10, 14–16; the rest of the problems are similar to others found here or in the homework (12 and 13 are complementary slackness problems).

3.10 The point of this problem is to see that while replacing v by $v_1 - v_2$ (when v can be positive or negative) with $v_1, v_2 \geq 0$ (needed for standard form) might look like we now have twice the work since one decision variable, v, has been replaced by two decision variables, v_1, v_2. However, v_1 and v_2 start out as nonbasic in the first dictionary, and we make the following claims:

(1) It is impossible for v_1, v_2 to be basic in any dictionary: for then A_B would have a v_1 column that is the same as the v_2 column except with signs reversed; in the case A_B would have these two columns summing to zero, and hence A_B would have a zero determinant, which is impossible.
(2) It is possible for v_1 to be basic and v_2 to be nonbasic: in this case v_2 only appears in the v_1 row, with a $+1$ cofficient (we saw this in class in two different ways), and so we can forget (at least temporarily) about v_2.
(3) It is possible for v_2 to be basic and v_1 to be nonbasic: this is similar to the last case.
(4) It is possible for v_1, v_2 to be nonbasic; in that case the coefficients of v_1 in the dictionary are the same as those for v_2 except the signs are reversed. So once we know the coefficients for v_1, we automatically get those for v_2.

In all cases figuring out how v_1 and v_2 are involved in a dictionary really only requires doing so for one of the variables. Hence this is only slightly more work than if we had v instead of v_1 and v_2; this is hardly twice the work.

A similar remark is true for an equality such as $p_1 + p_2 = 1$ which must be replaced by two inequalities in standard form. Although we get two slack variables,

$$w_1 = 1 - p_1 - p_2 \quad \text{and} \quad w_2 = -1 + p_1 + p_2$$

(instead of one slack variable), again it is not really twice the work. We see, for analogous reasons as in the previous paragraph, that w_1 and w_2 cannot both be nonbasic; furthermore, since $w_1 = -w_2$, when both w_1 and w_2 basic, then the rows for w_1 and w_2 are the same except with opposite signs; and when w_1 is basic and w_2 nonbasic (and similarly for vice versa), the row for w_1 must simply read: $w_1 = -w_2$.

3.12 The dual linear program to

$$\begin{aligned} \max 8x_{11} + 5x_{12} + 3x_{21} + 9x_{22} \quad \text{subject to} \\ x_{11} + x_{12} \le 1, \\ x_{21} + x_{22} \le 1, \\ x_{11} + x_{21} \le 1, \\ x_{12} + x_{22} \le 1, \\ x_{11}, x_{12}, x_{21}, x_{22} \ge 0 \end{aligned}$$

has dual decisions variables $y_{1,row}, y_{2,row}, y_{1,col}, y_{2,col}$ for the constraints, in order, and has constraints

$$\begin{aligned} 8 &\le y_{1,row} + y_{1,col}, \\ 5 &\le y_{1,row} + y_{2,col}, \\ 3 &\le y_{2,row} + y_{1,col}, \\ 9 &\le y_{2,row} + y_{2,col} \end{aligned}$$

which respectively correspond to $x_{11}, x_{12}, x_{21}, x_{22}$. Since the primal slack variables are all zero in optimality, we cannot draw any direct conclusions about the dual decision (y) variables. Since $x_{12} = x_{21} = 1$ in this proposed solution, we can conclude that

$$5 = y_{1,row} + y_{2,col} \quad \text{and} \quad 3 = y_{2,row} + y_{1,col}.$$

There are many solutions to these equation. However any such solution cannot be (dual) feasible, since these equations require

$$8 = 5 + 3 = y_{1,row} + y_{2,col} + y_{2,row} + y_{1,col},$$

but in feasibility we must also have

$$\begin{aligned} 8 &\le y_{1,row} + y_{1,col}, \\ 9 &\le y_{2,row} + y_{2,col}, \end{aligned}$$

and hence

$$17 \le y_{1,row} + y_{1,col} + y_{2,row} + y_{2,col}$$

which contradicts the fact that the right-hand-side must equal 8.

Hence, although complementary slackness gives us infinitely many possible values for the dual decision and dual slack variables, none of them can be feasible.

3.14 Consider the matrix game:

$$A = \begin{bmatrix} 1 & 36 \\ 4 & 25 \\ 9 & 16 \\ 16 & 9 \\ 25 & 4 \\ 36 & 1 \end{bmatrix}$$

(1) Show that for any $i = 2, 3, 4, 5$ we have that playing 50% of row $i-1$ and 50% of row $i+1$ is better than playing row i.

Answer: For $i = 2$ this amounts to

$$(.5)[1\ 36] + (.5)[9\ 16] = [5\ 26] > [4\ 25].$$

For other i we do a similar calculation.

(2) Show that if $\vec{f}(b)$ denotes the b-th row of this game, then $\vec{f}(b)$ is a convex (i.e., concave up) function of b.

Answer: We see that

$$\vec{f}(b) = [b^2\ (7-b)^2] = [b^2\ 49 - 14b + b^2].$$

Since both entries are b^2 plus linear functions of b, we have

$$\frac{d}{db}\vec{f}(b) = \vec{f}''(b) = [2\ 2]$$

which is positive, of $\vec{f}(b)$ is convex (concave up).

(3) What two rows will Alice play in "Alice plays a mixed strategy," based on your answers to the previous parts?

Answer Since $\vec{f}(b)$ is convex, the Alice will play some combination of the top and bottom rows.

(4) Find Betty's best mixed strategy.

Answer From the previous part, Alice and Betty are essentially playing the matrix game

$$\begin{bmatrix} 1 & 36 \\ 36 & 1 \end{bmatrix}$$

Since this matrix has a duality gap (of 35), we solve

$$\begin{bmatrix} 1 & 36 \\ 36 & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} v \\ v \end{bmatrix}, \qquad q_1 + q_2 = 1,$$

which yields $q_1 = q_2 = 1/2$.

3.15 Consider the matrix game:

$$A = \begin{bmatrix} -1 & -36 \\ -4 & -25 \\ -9 & -16 \\ -16 & -9 \\ -25 & -4 \\ -36 & -1 \end{bmatrix}$$

(1) Show that if $\vec{f}(b)$ denotes the b-th row of this game, then $\vec{f}(b)$ is a concave down function of b.

Answer: We see that

$$\vec{f}(b) = [-b^2\ \ -(7-b)^2] = [-b^2\ \ -49 + 14b - b^2].$$

Since both entries are $-b^2$ plus linear functions of b, we have

$$\frac{d}{db}\vec{f}(b) = \vec{f}''(b) = [-2 \ \ -2]$$

which is negative, and hence $\vec{f}(b)$ is concave down.

(2) What two rows will Alice play in "Alice plays a mixed strategy," based on your answer to the previous part?

Answer: Since $\vec{f}(b)$ is concave down, we know that Alice will play some mix of two adjacent row, and if the values of $\vec{f}(b)$ lie on both sides of the line $x = y$, then we choose the two b values that give $\vec{f}(b)$ closest to this line.

By inspection we can see that $[-9 \ \ -16]$ and $[-16 \ \ -9]$ are the two closest.

Alternatively, we can see that $\vec{f}(b)$ traces out a curve when b takes real values between 1 and 6, and $\vec{f}(b)$ intersects the line $x = y$ precisely when

$$-b^2 = -(7-b)^2$$

which happens precisely when $b = 3.5$. Hence the nearest integer b values, representing rows 3 and 4, are indicate the rows that Alice will play.

(3) Find Betty's best mixed strategy.

Answer From the previous part, Alice and Betty are essentially playing the matrix game

$$\begin{bmatrix} -9 & -16 \\ -16 & -9 \end{bmatrix},$$

and we similarly solve this 2×2 matrix game to see that Betty plays [1/2 1/2].

3.16 Explain each of the following principles of the revised simplex method (they are all discussed in this article):

(1) We do not need to compute the whole dictionary: assume we want to compute the entire ζ row to choose which variable enters, but then compute constants of the dictionary and only the column of an entering variable, which computations are involved? [Hint: look at Theorem 1.5.]

Answer: According to Theorem 1.5, we can compute the ζ row, and only the column of coefficients in $-A_B^{-1}A_N\vec{x}_N$ corresponding to a single entering variable x_i; hence we need only compute $A_B^{-1}A_i$, where A_i is the column of "big A" corresponding to x_i. (We also need to compute $A_B^{-1}\vec{b}$ or find it somehow.) So the real savings is in computing only the part of $-A_B^{-1}A_N\vec{x}_N$ corresponding to the entering variable x_i.

(2) The order of multiplcation matters: Theorem 1.5 has a term $\vec{c}_B^{\mathrm{T}} A_B^{-1} A_N$. In which order do we multiply these terms?

Answer: Since $\vec{c}_B$ is a row vector (of dimension $1 \times m$), we multiply this by A_B^{-1}, which again yields a row vector; this vector we multiply by A_N. The first multiplication of a $1 \times m$ row vector times an $m \times m$ matrix (when A_B^{-1} is a general matrix) takes time order m^2; the second multiplication (a $1 \times m$ times an $m \times n$ matrix) takes time order mn.

If we first multiplied A_B^{-1} times A_N, in general this would take time $m^2 n$, which would be much more expensive.

(3) How are eta matrices used in the revised simplex method? Specifically, consider the cost in terms of FLOPS (FLOating Point Operations): if to refactorize A_B^{-1} requires m^α FLOPS (where m is the number of basic variables), where α ranges between 1 and 3 depending on the problem, and each multiplication by an eta matrix requires m FLOPS, argue that to refactorize every s steps would involve an average of order $m^\alpha s^{-1}$ FLOPS (for the refactorization) and order ms FLOPS for the eta matrix multiplications. Then argue that the optimal value of s is $m^{(\alpha-1)/2}$.

Answer: To refactorize ever s steps means that the cost, m^α, is averaged over s steps, for m^α/s per step. Each step from $s/2$ to s involves at least $s/2$ Eta matrices, for a cost of at least $(s/4)m$ per step, since the Eta matrix multiplication takes order m FLOPS per step. The optimal value of order $sm + m^\alpha/s$ is achieved when these two terms balance, meaning that $sm = m^\alpha/s$, up to some constant multiple, so that s^2 is of order $m^{1+\alpha}$.

References

[Van14] Robert J. Vanderbei, *Linear programming: Foundations and extensions*, fourth ed., International Series in Operations Research & Management Science, 196, Springer, New York, 2014, Available to anyone with a UBC CWL account via http://link.springer.com.ezproxy.library.ubc.ca/book/10.1007%2F978-1-4614-7630-6. MR 3100219

Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, CANADA, and Department of Mathematics, University of British Columbia, Vancouver, BC V6T 1Z2, CANADA.

E-mail address: jf@cs.ubc.ca or jf@math.ubc.ca

URL: http://www.math.ubc.ca/~jf