February 2, 2000

possible outcomes are the same for the general duality scheme

		PRIMAL		
		optimal	**infeasible**	**unbounded**
DUAL	**optimal**	possible	impossible	impossible
	infeasible	impossible	possible	possible
	unbounded	impossible	possible	impossible

solving LPs in the form of the general duality scheme

a new algorithm is not needed because the general problem can easily be put into form for the simplex algorithm (see top of p. 142 in Chvatal, and also chapter 8).

algebra review

Consider an LP in form for the simplex algorithm: min c'x subject to Ax=b, $x \geq 0$. If B represents a choice of basic columns of A in some order, and N represents the nonbasic variables in some order, then we can write the equation Ax=b as $A_N x_N + A_B x_B = b$.

Solving for x_B, we get

$$x_B = A_B^{-1} b - A_B^{-1} A_N x_N.$$

For example, if $A = \begin{pmatrix} 1&1&1&0&0 \\ 1&0&0&1&0 \\ 0&1&0&0&1 \end{pmatrix}$, $b = \begin{pmatrix} 6 \\ 3 \\ 5 \end{pmatrix}$, B={2,4,1}, and N={5,3}, then $A_B = \begin{pmatrix} 1&0&1 \\ 0&1&1 \\ 1&0&0 \end{pmatrix}$, and $A_N = \begin{pmatrix} 0&1 \\ 0&0 \\ 1&0 \end{pmatrix}$. Setting x_N=0 and solving for x_B, we would have $x_B^* = A_B^{-1} b = \begin{pmatrix} 5 \\ 2 \\ 1 \end{pmatrix} \begin{matrix} \underline{2} \\ \underline{4} \\ \underline{1} \end{matrix}$ (We have written te basis on the side because without knowing the basis, it really has no useful meaning).

The objective function can be rewritten:

$$\begin{aligned} c'x &= c_B' x_B + c_N' x_N \\ &= c_B'(A_B^{-1} b - A_B^{-1} A_N x_N) + c_N' x_N \\ &= c_B' A_B^{-1} b + (c_N' - c_B' A_B^{-1} A_N) x_N. \end{aligned}$$

In this way, we obtain an abstract representation of the dictionary:

$$\begin{array}{l} x_B = A_B^{-1} b - A_B^{-1} A_N x_N \\ \hline z = c_B' A_B^{-1} b + (c_N' - c_B' A_B^{-1} A_N) x_N \end{array}$$

or of the tableau:

$$\begin{array}{l|l} x_B + A_B^{-1} A_N x_N & A_B^{-1} b \\ \hline \end{array}$$

$(c_N'-c_B'A_B^{-1}A_N)x_N$	$-c_B'A_B^{-1}b$

revised simplex method

This is not a new algorithm, just a more efficient implementation. It executes the algorithm without updating the entire dictionary (or tableau).

This method can be used even in situations where the columns of A are unknown (hard to imagine right now, but you will soon understand how this can be).

Summary of method:

0. We start each iteration knowing only $x_B^* = A_B^{-1}b$ and B.
1. We solve $y' = c_B'A_B^{-1}$ for y. Equivalently, we solve $y'A_B = c_B'$.
2. One by one, we compute the components of $c_N'-y'A_N$ until we find one which is positive. If none is found, STOP because we have found an optimal solution. If we do find one, we that nonbasic variable as the entering variable.
3. Let a denote the entering column (it is the column of A_N that gave us a positive number in step 3). Compute $d = A_B^{-1}a$ (by solving Bd=a, of course).
4. Find the largest $t \geq 0$ such that $x_B^* - td \geq 0$. If there is none, then STOP because the problem is unbounded. Otherwise, use the component that hits zero to determine the leaving variable.
5. Update basis: $x_B^* = x_B^* - td$.

example:

Consider the problem: max $x_1 + 2x_2$ subject to $x_1 + x_2 \leq 6$, $x_1 \leq 3$, $x_2 \leq 5$, $x_1, x_2 \geq 0$. If we were using the ordinary simplex method, our starting dictionary would be

$$\begin{aligned} x_3 &= 6 - x_1 - x_2 \\ x_4 &= 3 - x_1 \\ x_5 &= 5 - x_2 \\ z &= \phantom{6 - {}} x_1 + 2x_2 \end{aligned}$$

Instead, we start with $x_B^* = \begin{pmatrix} 6 \\ 3 \\ 5 \end{pmatrix} \begin{matrix} \underline{3} \\ \underline{4} \\ \underline{5} \end{matrix}$. Note that

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \quad c' = (1\ 2\ 0\ 0\ 0)$$

and that A and c' never change during the execution of the algorithm.

Iteration 1:

step 1: Solve $y'A_B = c_B'$ for y. Here, A_B contains columns 3,4,5 of A and c_B contains the 3,4,5 entries of c. Thus $A_B = I$ is the identity matrix and $c_B' = (0,0,0)$. Thus $y'=(0,0,0)$.

step 2: $c_N'-y'A_N = (1\ 2) - (0\ 0\ 0)\begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = (1\ 2)$. The columns of A_N are columns 1 and 2 of A. Since both entries of the result, (1 2), are positive, we can use either column 1 or 2 as the entering column. We will use column 1: $a = \begin{pmatrix}1\\1\\0\end{pmatrix}$, which corresponds to x_1 is the entering variable (or column).

step 3: Compute $d = A_B^{-1}a$ by solving $A_B d=a$. Since $A_B=I$, we have $d=a=\begin{pmatrix}1\\1\\0\end{pmatrix}$.

step 4&5: What multiple of d can be subtracted from x_B^* without producing any negative entries?

$$x_B^* - td = \begin{pmatrix}6\\3\\5\end{pmatrix} - t\begin{pmatrix}1\\1\\0\end{pmatrix} \geq \mathbf{0}.$$

Answer: t* = 3 is the largest multiple that can be subtracted. The result is $\begin{pmatrix}3\\0\\5\end{pmatrix}\begin{matrix}\underline{3}\\\underline{4}\\\underline{5}\end{matrix}$, and this is the new x_B^*, except that an entry that becomes zero is replaced by t*=3. This also identifies the leaving variable as x_4. Thus the new $x_B^* = \begin{pmatrix}3\\3\\5\end{pmatrix}\begin{matrix}\underline{3}\\\underline{1}\\\underline{5}\end{matrix}$.

Iteration 2:

We start with $x_B^* = \begin{pmatrix}3\\3\\5\end{pmatrix}\begin{matrix}\underline{3}\\\underline{1}\\\underline{5}\end{matrix}$.

step 1: Solve $y'A_B = c_B'$ for y. Here, A_B contains columns 3,1,5 of A and c_B contains the 3,1,5 entries of c. Thus $A_B = \begin{pmatrix}1&1&0\\0&1&0\\0&0&1\end{pmatrix}$ and $c_B' = (0,1,0)$. Thus $y'=(0,1,0)$.

step 2: $c_N'-y'A_N = (2\ 0) - (0\ 1\ 0)\begin{pmatrix}1&0\\0&1\\1&0\end{pmatrix} = (2\ -1)$. The columns of A_N are columns 2 and 4 of A. Since the first entry of the result, (2 -1), is positive, we can use either column 2 as the entering column: $a = \begin{pmatrix}1\\0\\1\end{pmatrix}$, which corresponds to x_2 is the entering variable (or column).

step 3: Compute $d = A_B^{-1}a$ by solving $A_B d=a$ or $\begin{pmatrix}1&1&0\\0&1&0\\0&0&1\end{pmatrix} d = \begin{pmatrix}1\\0\\1\end{pmatrix}$ which has the solution $d = \begin{pmatrix}1\\0\\1\end{pmatrix}$.

step 4&5: What multiple of d can be subtracted from x_B^* without producing any negative entries?

$$x_B^* - td = \begin{pmatrix}3\\3\\5\end{pmatrix} - t\begin{pmatrix}1\\0\\1\end{pmatrix} \geq \mathbf{0}.$$

Answer: $t^* = 3$ is the largest multiple that can be subtracted. The result is $\begin{pmatrix}0\\3\\2\end{pmatrix}\begin{matrix}\underline{3}\\\underline{1}\\\underline{5}\end{matrix}$, and this is the new $x_B{}^*$, except that an entry that becomes zero is replaced by $t^*=3$. This also identifies the leaving variable as x_3. Thus the new $x_B{}^* = \begin{pmatrix}3\\3\\2\end{pmatrix}\begin{matrix}\underline{2}\\\underline{1}\\\underline{5}\end{matrix}$.

Iteration 3:

We start with $x_B{}^* = \begin{pmatrix}3\\3\\2\end{pmatrix}\begin{matrix}\underline{2}\\\underline{1}\\\underline{5}\end{matrix}$.

step 1: Solve $y'A_B = c_B'$ for y. Here, A_B contains columns 2,1,5 of A and c_B contains the 2,1,5 entries of c. Thus $A_B = \begin{pmatrix}1&1&0\\0&1&0\\1&0&1\end{pmatrix}$ and $c_B' = (2\ 1\ 0)$. Thus $y'=(2\ {-1}\ 0)$.

step 2: $c_N'-y'A_N = (0\ 0) - (2\ {-1}\ 0)\begin{pmatrix}1&0\\0&1\\0&0\end{pmatrix} = (-2\ 1)$. The columns of A_N are columns 3 and 4 of A. Since the second entry of the result, $(-2\ 1)$, is positive, we can use either column 4 as the entering column: $a = \begin{pmatrix}0\\1\\0\end{pmatrix}$, which corresponds to x_4 as the entering variable (or column).

step 3: Compute $d = A_B{}^{-1}a$ by solving $A_Bd=a$ or $\begin{pmatrix}1&1&0\\0&1&0\\1&0&1\end{pmatrix} d = \begin{pmatrix}0\\1\\0\end{pmatrix}$ which has the solution $d = \begin{pmatrix}-1\\1\\1\end{pmatrix}$.

step 4&5: What multiple of d can be subtracted from $x_B{}^*$ without producing any negative entries?

$$x_B{}^* - td = \begin{pmatrix}3\\3\\2\end{pmatrix} - t\begin{pmatrix}-1\\1\\1\end{pmatrix} \geq \mathbf{0}.$$

Answer: $t^* = 2$ is the largest multiple that can be subtracted. The result is $\begin{pmatrix}5\\1\\0\end{pmatrix}\begin{matrix}\underline{2}\\\underline{1}\\\underline{5}\end{matrix}$, and this is the new $x_B{}^*$, except that an entry that becomes zero is replaced by $t^*=2$. This also identifies the leaving variable as x_5. Thus the new $x_B{}^* = \begin{pmatrix}5\\1\\2\end{pmatrix}\begin{matrix}\underline{2}\\\underline{1}\\\underline{4}\end{matrix}$.

Iteration 4:

We start with $x_B{}^* = \begin{pmatrix}5\\1\\2\end{pmatrix}\begin{matrix}\underline{2}\\\underline{1}\\\underline{4}\end{matrix}$.

step 1: Solve $y'A_B = c_B'$ for y. Here, A_B contains columns 2,1,4 of A and

c_B contains the 2,1,4 entries of c. Thus $A_B = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ and $c_B' = (2\ 1\ 0)$.

Thus y'=(1 0 1).

step 2: $c_N' - y'A_N = (0\ 0) - (1\ 0\ 1)\begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} = (-1\ -1)$. Since both entries are negative, the current $x_B^* = \begin{pmatrix} 5 \\ 1 \\ 2 \end{pmatrix}$ $\begin{matrix} \underline{2} \\ \underline{1} \\ \underline{4} \end{matrix}$ is optimal. The maximum value occurs at x=(1 5 0 2 0). The optimal value is c'x = 11.

HW 19: Suppose we wish to solve this problem by the revised simplex algorithm:

$$\max\ x_3 \text{ subject to}$$
$$-3x_1 - x_2 + x_3 \le 0$$
$$-2x_1 - 5x_2 + x_3 \le 0$$
$$x_1 + x_2 = 1$$
$$x_1 \ge 0,\ x_2 \ge 0.$$

(Note relationship to HW#18)

As a first step, let's introduce slack variables and replace the free variable x_3 with $x_3 - x_4$ with $x_3, x_4 \ge 0$. The constraints are now in the form Ax=b:

$$\max\ x_3 - x_4 \text{ subject to}$$
$$-3x_1 - x_2 + x_3 - x_4 + x_5 = 0$$
$$-2x_1 - 5x_2 + x_3 - x_4 + x_6 = 0$$
$$x_1 + x_2 = 1$$
$$\text{all variables} \ge 0$$

or the tableau

-3	-1	1	-1	1	0	0
-2	-5	1	-1	0	1	0
1	1	0	0	0	0	1
0	0	1	-1	0	0	0

By pivoting on the 3,1-entry you get a basic feasible solution that you can use to start the simplex algorithm:

0	2	1	-1	1	0	3	5
0	-3	1	-1	0	1	2	6
1	1	0	0	0	0	1	1
0	0	1	-1	0	0	0	

The simplex algorithm produces this sequence of iterates:

```
tableau:((0  5 0  0 1 -1  1)              basis=(5 3 1)
         (0 -3 1 -1 0  1  2)
         (1  1 0  0 0  0  1)
         (0  3 0  0 0 -1 -2))

tableau: ((0 1 0  0  1/5 -1/5   1/5)        basis=(2 3 1)
```

```
(0 0 1 -1  3/5  2/5  13/5)
(1 0 0  0 -1/5  1/5   4/5)
(0 0 0  0 -3/5 -2/5 -13/5))
```

Solve instead by using the revised simplex algorithm.

February 4, 2000

Exam #1 covering up to, but not including, revised simplex algorithm