Linear Programming & the Simplex Algorithm

Part II

P&S Chapter 2

Last Revised – October 7, 2006

Outline

- Moving from BFS to BFS.

- Organizing information in a tableau.

- How to move from a BFS to a *better* BFS and proving that an optimal BFS exists.

- The Simplex Algorithm.

Moving From BFS to BFS

For LP in standard form with matrix A, let x_0 be a BFS corresponding to the basis columns

$$\mathcal{B} = \{A_{B(i)} \ : \ i = 1, \ldots, m\}.$$

Let the basic components of x_0 be $x_{i,0}, i = 1, \ldots, m$, i.e.,

$$\sum_{i=1}^{m} x_{i,0} A_{B(i)} = b, \quad \text{where } x_{i,0} \geq 0.$$

Any nonbasic column, $A_j \in R^m$, $A_j \notin \mathcal{B}$ can be written as

$$\sum_{i=1}^{m} x_{i,j} A_{B(i)} = A_j.$$

Multiplying the 2nd by $\theta > 0$ and subtracting from the first gives

$$\boxed{\mathbf{\sum_{i=1}^{m} \left(x_{i,0} - \theta x_{i,j}\right) A_{B(i)} + \theta A_j = b}}$$

Given basis

$$\mathcal{B} = \{A_{B(i)} : i = 1, \ldots, m\}$$

we have seen that for every non-basis column $A_j \notin \mathcal{B}$

$$\boxed{\sum_{i=1}^{m} \left(\mathbf{x}_{i,0} - \theta \mathbf{x}_{i,j} \right) \mathbf{A}_{B(i)} + \theta \mathbf{A}_j = \mathbf{b}}$$

Assuming for the moment that x_0 is nondegenerate, so all $x_{i,0} > 0$,

Fixing j, start at $\theta = 0$ and then increase θ. As soon as $\theta > 0$ we have moved from a BFS to a feasible solution with $m + 1$ positive components.

How long does this solution remain feasible?
As long as $x_{i,0} - \theta x_{i,j} \geq 0$, i.e.,

$$\theta_0 = \min_{\substack{i \text{ s.t.} \\ x_{i,j} > 0}} \frac{x_{i,0}}{x_{i,j}}$$

$$\boxed{\sum_{i=1}^{m} \left(\mathbf{x_{i,0}} - \theta \mathbf{x_{i,j}}\right) \mathbf{A_{B(i)}} + \theta \mathbf{A_j} = \mathbf{b}}$$

$$\theta_0 = \min_{\substack{i \text{ s.t.} \\ x_{i,j} > 0}} \frac{x_{i,0}}{x_{i,j}}$$

There are two special cases:

a) x_0 is degenerate because some $x_{i,0} = 0$ and corresponding $x_{i,j} > 0$. Then $\theta_0 = 0$ and we do not move at all in R^n.

We actually stay at the same vertex but can think of what happened as moving to a new BFS in the LP, *representing the same vetex*, with column j replacing column $B(i)$,

In this case we sometimes say that x_j entered the basis at level 0.

b) If all the $x_{i,j}, i = 1, \ldots, m$ are nonpositive we would be able to move arbitrarily far without becoming infeasible. This would mean that F is unbounded, violating Assumption 3.

Theorem: Let x_0 be a BFS with basic components $x_{i,0},, i = 1, \ldots, m$ and basis
$\mathcal{B} = \{A_{B(i)} \ : \ i = 1, \ldots, m\}$.
Let j be s.t. $A_j \notin \mathcal{B}$.
Then the new feasible solution determined by

$$\theta_0 = \min_{\substack{i \text{ s.t.} \\ x_{i,j} > 0}} \frac{x_{i,0}}{x_{i,j}} = \frac{x_{l,0}}{x_{l,j}}$$

$$x'_{i,0} = \begin{cases} x_{i,0} - \theta_0 x_{i,j} & i \neq l \\ \theta_0 & i = l \end{cases}$$

is a BFS with basis $\mathcal{B}'$ defined by

$$B'(i) = \begin{cases} B(i) & i \neq l \\ j & i = l \end{cases}$$

When there is a tie in the min operation then the new BFS is degenerate

Proof: We already saw that this solution is feasible.

We now must show that it is basic, i.e, that the set of basis columns $\mathcal{B}'$ is *linearly independent*

Suppse not, then for some constants d_i, we have

$$0 = \sum_{i=1}^{m} d_i A_{B'(i)} = d_l A_j + \sum_{\substack{i=1 \\ i \neq l}}^{m} d_i A_{B'(i)}$$

Plugging in (why does this exist)

$$A_j = \sum_{i=1}^{m} x_{i,j} A_{B(i)}$$

gives

$$\sum_{\substack{i=1 \\ i \neq l}}^{m} \left(d_l x_{i,j} + d_i \right) A_{B'(i)} + d_l x_{l,j} A_{B(l)} = 0$$

Since $\mathcal{B}$ is a basis *all* of these coefficients must be zero so, in particular, $d_l x_{l,j} = 0$ so $d_l = 0$. But then, from the first equality (why) *all* of the $d_i = 0$ so the new basis is linearly independent.

If there is a tie then more than one of the components of x_0 become 0 and the basis is degenerate.

We just saw how to move from one BFS to another by removing one column $B(l)$ out of the basis and replacing it by another column. This method is called *pivoting*. Column $B(l)$ *leaves* the basis and new column j *enters* the basis.

Geometrically (to be proven later), a pivot either

a) moves from one vertex to another along an edge or

b) does nothing, i.e., stays at the same vertex. In this case, the corresponding BFSs must be degenerate.

We now see how to organize the equation information to make it easy to recognize and calculate pivots. In particular, we will see how to maintain the $x_{i,j}$ information in a tableau.

Organization of a Tableau

$$\begin{array}{rcrcrcrcl} 3x_1 & + & 2x_2 & + & x_3 & & & = & 1 \\ 5x_1 & + & x_2 & + & x_3 & + & x_4 & = & 3 \\ 2x_1 & + & 5x_2 & + & x_3 & + & x_5 & = & 4 \end{array}$$

We will keep a set of m equations in n unknowns in an $m \times n$ tableau:

	x_1	x_2	x_3	x_4	x_5
1	3	2	1	0	0
3	5	1	1	1	0
4	2	5	1	0	1

Note that the RHS of the equations has now become column 0 in tableau.

The *Elementary row operations*

- *Multiplying a row by a non-zero element*
- *Adding a multiple of one row to another row*

do not change the solutions to the set of equations. We can therefore use elementary row operations to manipulate the rows until a set of (basis) columns becomes an identity matrix.

As an example we can manipulate

$$\begin{array}{rcrcrcrcrcl} 3x_1 & + & 2x_2 & + & x_3 & & & & & = & 1 \\ 5x_1 & + & x_2 & + & x_3 & + & x_4 & & & = & 3 \\ 2x_1 & + & 5x_2 & + & x_3 & + & & + & x_5 & = & 4 \end{array}$$

so that it becomes

$$\begin{array}{rcrcrcrcrcl} 3x_1 & + & 2x_2 & + & x_3 & & & & & = & 1 \\ 2x_1 & - & x_2 & & & + & x_4 & & & = & 2 \\ -x_1 & + & 3x_2 & & & & & + & x_5 & = & 3 \end{array}$$

The manipulation on the previous page can also be performed in tableau form with

	x_1	x_2	x_3	x_4	x_5
1	3	2	1	0	0
3	5	1	1	1	0
4	2	5	1	0	1

becoming

	x_1	x_2	x_3	x_4	x_5
1	3	2	1	0	0
2	2	-1	0	1	0
3	-1	3	0	0	1

In this example the basis $\mathcal{B} = \{A_3, A_4, A_5\}$.

The important things to notice are that

- Column 0 gives the values of the basic variables $x_{B(i)} = x_{i,0}$, $i = 1, \ldots, m$ and
- The non basic columns contain exactly the values $x_{i,j}$ s.t. $A_j = \sum_i x_{i,j} A_{B(i)}$.

  Example: $A_1 = 3A_3 + 2A_4 - A_5 = \sum_{i=1}^{m} x_{i1}\, A_{B(i)}$

Current basis is $\mathcal{B} = \{A_3, A_4, A_5\}$. Suppose we want to move column $j = 1$ into basis.

	x_1	x_2	x_3	x_4	x_5
1	$\textcircled{3}$	2	1	0	0
2	2	-1	0	1	0
3	-1	3	0	0	1

Then

$$\theta_0 = \min_{\substack{i \\ \text{s.t. } x_{ij} > 0}} \left(\frac{x_{i0}}{x_{ij}} \right) = \frac{1}{3} \quad \text{for } i = l = 1$$

This means that we will introduce column A_1 into basis with the “1” in row $l = 1$. Doing this by elementary row operations gives a new basis $\mathcal{B}' = \{A_1, A_4, A_5\}$ and tableau

	x_1	x_2	x_3	x_4	x_5
$1/3$	1	$2/3$	$1/3$	0	0
$4/3$	0	$-7/3$	$-2/3$	1	0
$10/3$	0	$11/3$	$1/3$	0	1

Old

	x_1	x_2	x_3	x_4	x_5
1	$\boxed{3}$	2	1	0	0
2	2	-1	0	1	0
3	-1	3	0	0	1

Setting (col) $j = 1$ gives $\theta_1 = 1/3$, $i = l = 1$, and

	x_1	x_2	x_3	x_4	x_5
$1/3$	1	$2/3$	$1/3$	0	0
$4/3$	0	$-7/3$	$-2/3$	1	0
$10/3$	0	$11/3$	$1/3$	0	1

In general, if $x_{i,j}$ and $x'_{i,j}$ are the old and new variables, $\mathcal{B}$, $\mathcal{B}'$ the old and new bases and *pivot* is $x_{l,j}$ then the elementary row operations inserting column j into basis can be written as

$$\begin{array}{rcll} x'_{lq} & = & \dfrac{x_{lq}}{x_{lj}} & q = 0, \ldots, n \\ x'_{iq} & = & x_{iq} - x'_{lq} x_{ij} & i = 1, \ldots, m;\ i \neq l \\ & & & q = 0, \ldots n \\ B'(i) & = & \begin{cases} B(i) & i \neq l \\ j & i = l \end{cases} & \end{array}$$

Important: Basis remains *feasible* by definition of θ_1.

So far we have seen

- How to move from one BFS $\mathcal{B}$ to another BFS $\mathcal{B}'$ by moving a new column A_j into the basis set

- How to efficiently implement the above using a *tableau* representation

  - Given new column A_j, tableau permits calculating which old column $A_{B(i)}$ should be thrown out (this will be the i found when calculating θ_0).

  - Given i, j, tableau permits modifying old $x_{l,q}$ into new $x'_{l,q}$

It remains to show how to maintain cost in the tableau. This will be done introducing a new row into the tableau.

Before doing this we first show that there is a simple criteria to decide whether a BFS is optimal.

This criteria will have the added advantage of implying that there is always an optimal BFS.

Cost of BFS x_0 with basis $\mathcal{B}$ is $z_0 = \sum_{i=1}^{m} x_{i,0} c_{B(i)}$.

Before A_j is brought into basis we have

$$A_j = \sum_{i=1}^{m} x_{i,j} A_{B(i)}.$$

This means that for every unit of x_j that enters BFS, $x_{i,j}$ units of $x_{B(i)}$ must leave. A unit increase of x_j implies a net change in cost or *relative cost* of

$$\boxed{\bar{\mathbf{c}}_{\mathbf{j}} = \mathbf{c}_{\mathbf{j}} - \sum_{\mathbf{i}=1}^{\mathbf{m}} \mathbf{x}_{\mathbf{i},\mathbf{j}} \mathbf{c}_{\mathbf{B}(\mathbf{i})} = \mathbf{c}_{\mathbf{j}} - \mathbf{z}_{\mathbf{j}}}$$

where $z_j = \sum_{i=1}^{m} x_{i,j} c_{B(i)}$.
It is therefore *profitable* to bring column j into basis iff $\bar{c}_j < 0$.
We will soon see that BFS is optimal iff $\forall j, \bar{c}_j \geq 0$.

Notation: For tableau X let B be the $m \times n$ matrix containing basis columns in X. Let c_B be the m-vector of costs corresponding to this basis. Let $z = col(z_1, \ldots, z_n)$ Then, since X comes from diagonalizing basis columns of A,

$$X = B^{-1}A \quad \text{and} \quad z' = c'_B X = c'_B B^{-1} A$$

Optimality Theorem:
At BFS x_0 a pivot step in which x_j enters the basis changes the cost by

$$\theta_0 \bar{c}_j = \theta_0(c_j - z_j)$$

Furthermore, if

$$\bar{c} = c - z \geq 0$$

then x_0 is optimal.

Proof: Recall that original cost was $z_0 = \sum_{i=1}^{m} x_{i,0} c_{B(i)}$. If j moves into basis we have already seen that

$$x'_{i,0} = \begin{cases} x_{i,0} - \theta_0 x_{i,j} & i \neq l \\ \theta_0 & i = l \end{cases}$$

so new cost is

$$\begin{aligned} z'_0 &= \sum_{i \neq l; i=1}^{m} (x_{i,0} - \theta_0 x_{i,j}) c_{B(i)} + \theta_0 c_j \\ &= z_0 + \theta_0 (c_j - z_j) \end{aligned}$$

proving first part. To prove second part let y be *any* feasible vector (not necessarily basic), i.e, $Ay = b$ and $y \geq 0$. Then

$$c'y \geq z'y = c'_B B^{-1} Ay = c'_B B^{-1} b = c' x_0$$

where last equality comes from fact that $b = Bx_0$.

Optimality Theorem:
At BFS x_0 a pivot step in which x_j enters the basis changes the cost by

$$\theta_0 \bar{c}_j = \theta_0 (c_j - z_j)$$

Furthermore, if

$$\bar{c} = c - z \geq 0$$

then x_0 is optimal.

This is the most important theorem we will see in this section!!

Consider the following algorithm:

Start at some BFS and loop the following line

If any $\bar{c}_j < 0$, pivot on A_j and construct new BFS.

(Ignoring degeneracy and assumptions we made) theorem implies that pivot always decreases cost, so cost is decreasing monotonically and we can never loop back to previously seen BFS. Since there are only a finite number of BFSs, the algorithm must terminate. The theorem then implies that the BFS at which we terminated was an optimal solution.

This algorithm is the **SIMPLEX ALGORITHM**.