ORIE 6300 Mathematical Programming I October 6, 2016

Lecture 14

Lecturer: Damek Davis *Scribe: Chamsi Hssaine*

1 Pivot Rules

A key factor in the performance of the simplex method is the rule we use to decide which j (s.t. $\bar{c}_j < 0$) should enter the basis on each pivot. We know, from the last lecture, that the time spent in checking $\bar{c}_j < 0$, for each j, is $O(m)$, and if we check all possible j's, the total time is $O(mn)$. This compares with the $O(km)$ time needed to complete the rest of the pivot, where k is the number of pivots performed since we last computed A_B^{-1}. However, the selection of a pivot rule not only will affect the performance of each pivot, but also the total number of pivots needed to reach the optimum (if it exists). The following are common practices on how to do this; however, remember that these methods are only heuristics. Below (+)=Advantage and (−)=Disadvantage.

1. Use first negative reduced cost $\bar{c}_j$ found.
   (+) It is computationally cheap.
   (−) We are not sure whether we are making progress or not.

2. Use most negative reduced cost $\bar{c}_j$ found.
   (−) It is more expensive than the first.
   (−) We are not sure of progress, since we could have $\varepsilon_j = 0$.

3. Use greatest improvement $\bar{c}_j\varepsilon$ found.
   (+) We are clearly making progress.
   (−) Needs even more computation. We need to compute: $A_B d = A_j$, $\forall j$ s.t. $\bar{c}_j < 0$.
   Then for each j compute: $\max \varepsilon_j$ s.t. $\varepsilon_j d \leq \bar{b} = x_B$ and set $\varepsilon = \text{argmax}_{\varepsilon_j} \bar{c}_j \varepsilon_j$

4. Use the steepest edge:
$$\min_j \ \frac{\bar{c}_j}{\|d\|_2}$$
   The reasoning for this method is that $d = A_B^{-1} A_j$ represents the vector of change.
$$\hat{x}_B = x_B - \bar{A}(\varepsilon e_j) = x_B - \varepsilon d$$
   Solution x_B moves in direction $-d$. Consequently, the steepest edge method gives the greatest objective value improvement per amount of current solution perturbation.
   (−) Computationally expensive. Less than "greatest improvement" (we only compute d for each j, not ε).
   (+) Works well in practice. Seems to lead to fewer overall pivots than greatest improvement.

2 Pivot Pool

For LP problems with many variables, the use of a "pivot pool" often works well. This is how a pivot pool works. Periodically, the algorithm computes the reduced costs for all the variables, and among those with $\bar{c}_j < 0$ a subset of "best" variables are chosen. This subset is used in all further iterations of the simplex method until the pivot pool either becomes empty or grows too old. This allows the algorithm to choose entering variables quickly (by one of rules (2), (3), or (4), above, for example), but only considering a tuned subset of the entire set of variables.

3 Cycling & Bland's Rule

Consider a linear programming problem input in standard form: $\min(c^T x : x \geq 0, Ax = b)$.
Recall: A basic feasible solution, x, is **degenerate** if $\exists\ i$ such that $x_i = 0$ and i is in the basis corresponding to x.
Possible Problem: If x is **degenerate** and $i \in B$, s.t. $x_i = 0$, is chosen to leave the basis, then the objective function doesn't change.

Another Problem: Degeneracy can potentially cause cycling in the simplex method.

Definition 1 *A **cycle** in the simplex method is a sequence of $\kappa + 1$ iterations with corresponding bases $B_0,\ldots,B_\kappa,B_0$ and $\kappa \geq 1$.*

This is not a theoretical concern; this can actually happen. In Example 1, a sequence of pivots leads back to the initial basis (i.e. cycling occurs).

Example 1 *(Degenerate Pivoting)*
Pivot rules:

- *Choose entering variable with largest reduced cost.*
- *Choose leaving variable with smallest subscript.*

$$
\begin{array}{lrcrcrcrcl}
\textit{maximize} & 10x_1 & - & 57x_2 & - & 9x_3 & - & 24x_4 & & \\
\textit{subject to} & 0.5x_1 & - & 5.5x_2 & - & 2.5x_3 & + & 9x_4 & \leq & 0 \\
 & 0.5x_1 & - & 1.5x_2 & - & 0.5x_3 & + & x_4 & \leq & 0 \\
 & x_1 & & & & & & & \leq & 1 \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & \geq & 0.
\end{array}
$$

Introduce slacks in initial tableau. Initial basis: $\{5, 6, 7\}$.

$$
\begin{array}{lrrrrrrrcl}
z & -10x_1 & +57x_2 & +9x_3 & +24x_4 & & & & = & 0 \\
 & \boxed{0.5x_1} & -5.5x_2 & -2.5x_3 & +9x_4 & +x_5 & & & = & 0 \\
 & 0.5x_1 & -1.5x_2 & -0.5x_3 & +x_4 & & +x_6 & & = & 0 \\
 & x_1 & & & & & & +x_7 & = & 1.
\end{array}
$$

$$
\begin{array}{rrrrrrrcl}
z & & -53x_2 & -41x_3 & +204x_4 & +20x_5 & & & = 0 \\
 & x_1 & -11x_2 & -5x_3 & +18x_4 & +2x_5 & & & = 0 \\
 & & \boxed{4x_2} & +2x_3 & -8x_4 & -x_5 & +x_6 & & = 0 \\
 & & 11x_2 & +5x_3 & -18x_4 & -2x_5 & & +x_7 & = 1.
\end{array}
$$

$$
\begin{array}{rrrrrrrcl}
z & & & -14.5x_3 & +98x_4 & +6.75x_5 & +13.25x_6 & & = 0 \\
 & x_1 & & \boxed{+0.5x_3} & -4x_4 & -0.75x_5 & +2.75x_6 & & = 0 \\
 & & x_2 & +0.5x_3 & -2x_4 & -0.25x_5 & +0.25x_6 & & = 0 \\
 & & & -0.5x_3 & +4x_4 & +0.75x_5 & -2.75x_6 & +x_7 & = 1.
\end{array}
$$

$$
\begin{array}{rrrrrrrcl}
z & +29x_1 & & & -18x_4 & -15x_5 & +93x_6 & & = 0 \\
 & 2x_1 & & +x_3 & -8x_4 & -1.5x_5 & +5.5x_6 & & = 0 \\
 & -x_1 & +x_2 & & \boxed{+2x_4} & +0.5x_5 & -2.5x_6 & & = 0 \\
 & x_1 & & & & & & +x_7 & = 1.
\end{array}
$$

$$
\begin{array}{rrrrrrrcl}
z & +20x_1 & +9x_2 & & & -10.5x_5 & +70.5x_6 & & = 0 \\
 & -2x_1 & +4x_2 & +x_3 & & \boxed{+0.5x_5} & -4.5x_6 & & = 0 \\
 & -0.5x_1 & +0.5x_2 & & +x_4 & +0.25x_5 & -1.25x_6 & & = 0 \\
 & x_1 & & & & & & +x_7 & = 1.
\end{array}
$$

$$
\begin{array}{rrrrrrrcl}
z & -22x_1 & +93x_2 & +21x_3 & & & -24x_6 & & = 0 \\
 & -4x_1 & +8x_2 & +2x_3 & & +x_5 & -9x_6 & & = 0 \\
 & +0.5x_1 & -1.5x_2 & -0.5x_3 & +x_4 & & \boxed{+x_6} & & = 0 \\
 & x_1 & & & & & & +x_7 & = 1.
\end{array}
$$

New basis, $\{5, 6, 7\}$*, is identical with the first basis, so now we* ***CYCLE!***

But this solution has objective value 0*, so is* ***NOT*** *optimal:* $x = (1, 0, 1, 0)$ *is better.*

There are a couple of ways for avoiding cycling. One way is the "Perturbation Method". In this, we perturb the right-hand side b by small amounts different for each b_i. Then, $x_i \neq 0 \ \forall i \in B$ since x_i will always have some linear combination of these small amounts. See the textbook for details. This is equivalent to a lexicographic method for breaking ties in choosing entering and exiting variables.

Another way to avoid cycling is a lexicographic method known as Bland's Rule.

Definition 2 (Bland's Rule) *Choose the entering basic variable* x_j *such that* j *is the smallest index with* $\bar{c}_j < 0$*. Also choose the leaving basic variable* i *with the smallest index (in case of ties in the ratio test).*

Theorem 1 (Termination with Bland's Rule) *If the simplex method uses Bland's rule, it terminates in finite time with optimal solution. (i.e. no cycling)*

Proof: Suppose the simplex method is implemented with Bland's rule and a cycle exists. Then there exist bases $B_0,\ldots,B_\kappa,B_0$ that form the cycle. Additionally, recall that the objective value and the current solution x^* remain the same throughout the cycle. The solutions remain the same because $\hat{x}_B = x_B - \bar{A}(\epsilon e_j)$. Since $\epsilon = 0$, $\hat{x}_B = x_B$.

Definition 3 *A variable x_j is fickle if it is in some, but not all, bases B_i in the cycle.*

Clearly, if x_j^* is fickle, then x_j^*=0 throughout the cycle.
Let x_t be the fickle variable with the largest index. Then we know:

(1) There exists a basis B in the cycle such that $t \in B$, but is not in the next basis in the cycle. Let s be the index of entering variable. It must be the case that s is fickle because it wasn't in B, but enters the next basis in the cycle.

As before, we have $\bar{A} = A_B^{-1}A_N$, $\bar{b} = A_B^{-1}b$ and the reduced cost $\bar{c}$ s.t. $\bar{c}_B = 0$.

Note that since s enters and t leaves the basis, $\bar{A}_{st} > 0$. Since s is fickle, by definition of t, $s < t$. Also $\bar{c}_s < 0$ (s enters) and $\bar{c}_t = 0$ (since $t \in B$).

The direction of movement, d, in this case is $\begin{pmatrix} -\bar{A}_s \\ e_s \end{pmatrix}$ where $-\bar{A}_s$ corresponds to direction of movements for basic variables and e_s to nonbasic variables. The reason is that $\tilde{x}_B = \bar{b} - \epsilon\bar{A}_s$, $\tilde{x}_j = 0$ for all $j \in N - \{s\}$ and x_s increases by ϵ.

(2) There exists a basis $\hat{B}$ in the cycle such that $t \notin \hat{B}$ and t is selected to be the entering variable for the next basis. Let $\hat{c}$ denote the reduced costs corresponding to $\hat{B}$. Again, since t is entering the new basis, $\hat{c}_t < 0$. Also since s is fickle, by definition $s < t$ and by using the pivoting rule, either $s \in B$ which implies $\hat{c}_s = 0$ or $c \in N$ which implies $\hat{c}_s \geq 0$; in either case $\hat{c}_s \geq 0$. More generally for all fickle j such that $j \neq t$, $\hat{c}_j \geq 0$.

We summarize the facts about vector d in the following tabular form.

$d_t = -\bar{A}_{st} < 0$	
$d_s = 1 > 0$	
$d_j \geq 0$	if $j \in B$, $j \neq t$, j is fickle
	($d_j < 0 \Leftrightarrow \bar{A}_{sj} > 0$ because of the pivot rule)
$d_j = 0$	if $j \in N$, $j \neq s$
d_j =?	if $j \in B$, j is not fickle

Note that $Ad = 0$. ($Ad = A_B(-A_B^{-1}A_s) + A_N e_s = -A_s + A_s = 0$, or more intuitively, in order for the primal solution to stay feasible, the direction of movement is in the nullspace of A). Since $\bar{c}$ and $\hat{c}$ are vectors of reduced cost corresponding to B and $\hat{B}$ respectively, we have: $\bar{c} = c - A^T\bar{y}$ and $\hat{c} = c - A^T\hat{y}$. Define $\tilde{c} = \hat{c} - \bar{c} = A^T(y - \hat{y})$. Then $\tilde{c}$ is a linear combination of columns of A^T and therefore a linear combination of rows of A. On the other hand $Ad = 0$ implies d is orthogonal to all rows of A. Hence:

$$\tilde{c}^T d = 0$$

We complete the table by adding $\tilde{c}_j$ values for different j:

$d_t = -\bar{A}_{st} < 0$	$\tilde{c}_t = \hat{c}_t - \bar{c}_t < 0$	
$d_s = 1 > 0$	$\tilde{c}_s = \hat{c}_s - \bar{c}_s > 0$	
$d_j \geq 0$	$\tilde{c}_j = \hat{c}_j - \bar{c}_j \geq 0$	if $j \in B$, $j \neq t$, j is fickle
$d_j = 0$	$\tilde{c}_j =?$	if $j \in N$, $j \neq s$
$d_j =?$	$\tilde{c}_j = \hat{c}_j - \bar{c}_j = 0$	if $j \in B$, j is not fickle

$\tilde{c}^T d$ can also be computed using the values in the table:

$$0 = \tilde{c}^T d = \tilde{c}_s d_s + \tilde{c}_t d_t + \sum_{\substack{j \in B \\ \text{fickle} \\ j \neq t}} \tilde{c}_j d_j + \sum_{\substack{j \in B \\ \text{not fickle}}} \tilde{c}_j d_j + \sum_{\substack{j \in N \\ j \neq s}} \tilde{c}_j d_j > 0$$

Thus we have reached a contradiction.

□

Therefore, if we make our entering and leaving variables selections according to Bland's rule, we never cycle. Therefore in each iteration we have a unique basis and hence the simplex algorithm terminates after a finite number of iterations, since there are a finite number of possible bases.