

Lecture 13

Lecturer: David P. Williamson

Scribe: Enrique Mallada

1 Pivot Rules

A key factor in the performance of the simplex method is the rule we use to decide which j (st $\bar{c}_j < 0$) 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.
3. Use greatest improvement $\bar{c}_j \varepsilon$ found.
 - (+) We are clearly making progress.
 - (-) Needs even more computation. We need to compute: $A_B d = A_j, \forall j \text{ s.t. } \bar{c}_j < 0$. Then for each j compute: $\max \varepsilon_j \text{ s.t. } \varepsilon_j d \leq \bar{b} = x_B$ and set $\varepsilon = \arg \max_{\varepsilon_j} \bar{c}_j \varepsilon_j$
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.

$$\hat{x}_B = x_B - \bar{A}(\varepsilon e_j) = x_B - \varepsilon d$$

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: minimize $c^T x$ subject to $Ax = b$, $x \geq 0$. S

Recall: A basic feasible solution, x , is **degenerate** if $\exists i$ such that $\hat{x}_i = 0$ and i is in the basis corresponding to x . Alternatively, a basic feasible solution is degenerate if there exist different bases B_0, \dots, B_κ (with $\kappa \geq 1$) 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 $B_0, \dots, B_\kappa, B_0$ and $\kappa \geq 1$.

This isn’t a theoretical concern; this can actually happen; see the attached example. In our example, we used the following rules for pivot selection:

1. Choose a column whose reduced cost is the most negative to enter the basis.
2. Out of all variables eligible to exit, choose one with the smallest subscript.

As a result, a sequence of pivots leads back to the initial basis. (i.e. cycling occurs). See the example for details.

There are a couple of ways of 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 $\bar{c}_j < 0$ and $j = \min\{i : \bar{c}_i < 0\}$; i.e., the variable with the smallest index of all those eligible to enter. Choose the leaving basic variable, in the event of a tie, again with the smallest index.

Theorem 1 (Termination with Bland’s Rule) If the simplex method is implemented with Bland’s Rule, then it is guaranteed to terminate.

Proof: Suppose the simplex method is implemented with Bland’s rule and a cycle exists. Then there exist bases $B_0, \dots, B_\kappa, B_0$ that form the cycle. Additionally, recall that the objective value and the current solution x^* remain constant throughout the cycle.

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 such that $t \in B$, but is not in the basis for the next iteration. Let x_s be the entering basic variable in this iteration (for some s). 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. $\bar{c}_B = 0$ and $\bar{c}_N = c_N - A_N^T((A_B^T)^{-1}c_B)$. Note that, since s is entering the basis, $\bar{c}_s < 0$ and $\bar{A}_{st} > 0$. Further more, s is also fickle, and by definition of t , $s < t$.
- (2) There exists a basis \hat{B} 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 in this iteration. Again, since t is entering the new basis, $\hat{c}_t < 0$. And $\hat{c}_s \geq 0$, since if $s \notin \hat{B}$, then it must be the case where $\hat{c}_s \geq 0$ (o.w. wouldn't have chosen t), and if $s \in \hat{B}$, by definition of reduced cost $\hat{c}_s = 0$.

Recall that by our formulation of reduced costs, the objective function $c^T x$, for any x satisfying $Ax = b$ (not necessarily feasible), is offset from the reduced costs by some constant. Thus

$$c^T x = \bar{v} + \bar{c}^T x = \hat{v} + \hat{c}^T x,$$

where \bar{v} and \hat{v} are constants. Even more, since x^* satisfies $Ax^* = b$, $\bar{c}_B = \hat{c}_B = 0$ and $x_N^* = x_{\hat{N}}^* = 0$, we have $\bar{v} = \hat{v}$ since

$$\begin{aligned} \bar{c}^T x^* + \bar{v} &= \hat{c}^T x^* + \hat{v} \\ \Rightarrow \bar{c}_B^T x_B^* + \bar{c}_N^T x_N^* + \bar{v} &= \hat{c}_B^T x_B^* + \hat{c}_N^T x_N^* + \hat{v} \\ &\Rightarrow \bar{v} = \hat{v}. \end{aligned}$$

Finally, this implies $\bar{c}^T x = \hat{c}^T x$.

Now, consider the following family of (not necessarily basic or feasible) solutions that do satisfy $A\tilde{x} = b$: $\tilde{x}_s = \varepsilon$, and $\tilde{x}_B = \bar{b} - \varepsilon \bar{A}_s$ where $A_B^{-1}A_s = \bar{A}_s$. (It is very important to recall that the equivalence of the objective functions was with respect to any solution to $Ax = b$, and this did not require feasibility.) Now we can use our previous result for \tilde{x} . Then,

$$\begin{aligned} \bar{c}^T \tilde{x} &= \hat{c}^T \tilde{x} \\ \Rightarrow \bar{c}_B^T \tilde{x}_B + \bar{c}_N^T \tilde{x}_N &= \hat{c}_B^T \tilde{x}_B + \hat{c}_N^T \tilde{x}_N \\ \Rightarrow 0 + \bar{c}_s \varepsilon &= \hat{c}_B^T (\bar{b} - \varepsilon \bar{A}_s) + \hat{c}_s \varepsilon \end{aligned}$$

Since the last equality holds for any ε , then $\bar{c}_s = -\hat{c}_B^T \bar{A}_s + \hat{c}_s$. But, we know that $\bar{c}_s < 0$ and $\hat{c}_s \geq 0$, so it must be the case that $\hat{c}_B^T \bar{A}_s > 0$. Then there exists $r \in B$ s.t. $\hat{c}_r \bar{A}_{sr} > 0$, which implies $\hat{c}_r \neq 0$. Therefore, $r \notin \hat{B}$. However, $r \in B$, so r must be fickle and $r \leq t$.

We then have two cases:

Case 1: $r = t$. But, we know $\hat{c}_t < 0$ and $\bar{A}_{st} > 0$, then, $\hat{c}_t \bar{A}_{st} < 0$, which is a contradiction.

Case 2: $r < t$. Since we didn't choose r to enter \hat{B} , then $\hat{c}_r \geq 0$. Using $\hat{c}_r \bar{A}_{sr} > 0$, we get $\hat{c}_r > 0$ and $\bar{A}_{sr} > 0$. But, r is fickle, which implies $x_r^* = 0$ and $\bar{b}_r = 0$. By the ratio rule, r has a ratio of zero, and thus, r is a candidate to leave B , which is a contradiction since $r < t$ and we chose t to leave B .

□

Therefore, if we make our entering and leaving variables selections according to Bland's rule, we never create a cycle, and hence the simplex algorithm terminates after a finite number of iterations.