Reading for This Lecture

• Bertsimas 3.2-3.4
Linear Programming

- We consider solution of a *linear program* in standard form:

\[
\begin{align*}
\min & \quad c^\top x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

where \( A \in \mathbb{R}^{m \times n} \), \( c \in \mathbb{R}^n \), and \( b \in \mathbb{R}^m \).

- The most commonly used algorithm for solving this problem is the *simplex algorithm*. 
Implementing the Simplex Method

“Naive” Implementation

1. Start with a basic feasible solution \( \hat{x} \) with indices \( B(1), \ldots, B(m) \) corresponding to the current basic variables.

2. Form the basis matrix \( B \) and compute \( p^\top = c_B^\top B^{-1} \) by solving \( p^\top B = c_B^\top \).

3. Compute the reduced costs by the formula \( \bar{c}_j = c_j - p^\top A_j \). If \( \bar{c} \geq 0 \), then \( \hat{x} \) is optimal.

4. Select the entering variable \( j \) and obtain \( u = B^{-1}A_j \) by solving the system \( Bu = A_j \). If \( u \leq 0 \), the LP is unbounded.

5. Determine the step size \( \theta^* = \min\{i | u_i > 0\} \frac{\hat{x}_{B(i)}}{u_i} \).

6. Determine the new solution and the leaving variable \( i \).

7. Replace \( i \) with \( j \) in the list of basic variables.

8. Go to Step 1.
Calculating the Basis Inverse

- Note that most of the effort in each iteration of the Simplex algorithm is spent solving the systems

\[ p^\top B = c_B \]
\[ Bu = A_j \]

- If we knew \( B^{-1} \), we could solve both of these systems.

- Calculating \( B^{-1} \) quickly and accurately is the biggest challenge of implementing the simplex algorithm.

- The full details of how to do this are beyond the scope of this course.

- We will take a cursory look at these issues in the rest of the chapter.
**Efficiency of the Simplex Method**

- To judge efficiency, we calculate the number of arithmetic operations it takes to perform the algorithm.

- To solve a system of $m$ equations and $m$ unknowns, it takes *on the order of* $m^3$ operations, denoted $O(m^3)$.

- To take the inner product of two $n$-dimensional vectors takes $O(n)$ operations ($n$ multiplications and $n$ additions).

- Hence, each iteration of the naive implementation of the Simplex method takes $O(m^3 + mn)$ operations.

- We'll try to improve upon this.
Improving the Efficiency of Simplex

- Again, the matrix $B^{-1}$ plays a central role in the simplex method.
- If we had $B^{-1}$ available at the beginning of each iteration, we could easily compute everything we need.
- Recall that $B$ changes in only one column during each iteration.
- How does $B^{-1}$ change?
- It may change a lot, but we can update it instead of recomputing it.
Updating the Basis Inverse

• We have $B^{-1}B = I$, so that $B^{-1}A_{B(i)}$ is the $i$th unit vector $e_i$.

• If $B$ is the old basis and $\bar{B}$ is the new one, then

$$B^{-1}\bar{B} = \begin{bmatrix} e_1 & \cdots & e_{i-1} & u_{i+1} & \cdots & e_m \end{bmatrix}$$

$$= \begin{bmatrix} 1 & u_1 \\ \vdots & \vdots \\ u_l & \vdots \\ \vdots & \ddots \\ u_m & 1 \end{bmatrix}$$

• We want to turn this matrix into $I$ using elementary row operations.

• If we apply these same row operations to $B^{-1}$, we will turn it into $\bar{B}^{-1}$.
Representing Elementary Row Operations

- Performing an elementary row operation is the same as left-multiplying by a specially constructed matrix.

- To multiply the $j$th row by $\beta$ and add it to the $i$th row, take $I$ and change the $(i, j)$th entry to $\beta$.

- A sequence of row operations can similarly be represented as a matrix.

- Hence, we can change $B^{-1}$ into $\bar{B}^{-1}$ by left-multiplying by a matrix $Q$ which looks like

$$Q = \begin{bmatrix}
1 & \frac{-u_1}{u_l} \\
\vdots & \ddots \\
\frac{1}{u_l} & \ddots & \ddots \\
\frac{-u_m}{u_l} & \cdots & 1
\end{bmatrix}$$
The Revised Simplex Method

A typical iteration of the revised simplex method:

1. Start with a specified BFS $\hat{x}$ and the associated basis inverse $B^{-1}$.
2. Compute $p^\top = c_B B^{-1}$ and the reduced costs $\bar{c}_j = c_j - p^\top A_j$.
3. If $\bar{c} \geq 0$, then the current solution is optimal.
4. Select the entering variable $j$ and compute $u = B^{-1} A_j$.
5. If $u \leq 0$, then the LP is unbounded.
6. Determine the step size $\theta^* = \min\{i | u_i > 0\} \frac{\hat{x}_B(i)}{u_i}$.
7. Determine the new solution and the leaving variable $i$.
8. Update $B^{-1}$.
9. Go to Step 1.
Some Notes on the Simplex Method

• One key element not described above is how to construct an initial feasible basis.

• If we start with a feasible basis, each iteration of the simplex methods ends with a new basic feasible solution (assuming nondegeneracy).

• This is all we need to prove the following result:

**Theorem 1.** Consider a linear program over a nonempty polyhedron $\mathcal{P}$ and assume every basic feasible solution is nondegenerate. Then the simplex method terminates after a finite number of iterations in one of the following two conditions:

- We obtain an **optimal** basis and a corresponding optimal basic feasible solution.
- We obtain a vector $d \in \mathbb{R}^n$ such that $Ad = 0$, $d \geq 0$, and $c^\top d < 0$, and the LP is unbounded.
Pivot Selection

• The process of removing one variable and replacing from the basis and replacing it with another is called *pivoting*.

• We have the freedom to choose the entering variable from among a list of candidates.

• How do we make this choice?

• The reduced cost tells us the cost in the objective function for each unit of change in the given variable.

• Intuitively, $c_j$ is the cost for the change in the variable itself and $-c_B^T B^{-1} A_j$ is the cost of the compensating change in the other variables.

• This leads to the following possible rules:
  – Choose the column with the most negative reduced cost.
  – Choose the column for which $\theta^*|\bar{c}_j|$ is largest.
Other Pivoting Rules

• In practice, sophisticated pivoting rules are used.

• Most try to estimate the change in the objective function resulting from a particular choice of pivot.

• For large problems, we may not want to compute all the reduced costs.

• Remember that all we require is some variable with negative reduced cost.

• It is not necessary to calculate all of them.

• There are schemes that calculate only a small subset of the reduced costs each iteration.
Simplex for Degenerate Problems

- If the current BFS is degenerate, then the step size might be limited to zero (why?).
  - This means that the next feasible solution is the same as the last.
  - We can still form a new basis, however, as before.

- Even if the step-size is positive, we might end up with one or more basic variables at level zero.
  - In this case, we have to decide arbitrarily which variable to remove from the basis.
  - The new solution will be degenerate.

- Degeneracy can cause cycling, a condition in which the same feasible solution is reached more than once.

- If the algorithm doesn’t terminate, then it must cycle.
Anticycling and Bland’s Rule

• Bland’s pivoting rule:
  – The entering variable is the one with the smallest subscript among those whose reduced costs are negative.
  – The leaving variable is the one with the smallest subscript among those that are eligible to leave the basis.

• Bland’s rule guarantees that cycling cannot occur.

• We also don’t need to compute all the reduced costs.
Numerical Considerations

• In the simplex algorithm, we are solving a sequence of closely related systems of equations.

• The factorization we are using to solve each of these systems is updated and round-off error accumulates.

• In practice, it is common to periodically discard the basis factorization and re-compute it from scratch to combat this problem.

• What factors affect the accuracy of solving just one of these systems from scratch?

• Naturally, the condition number of the current basis is important.

• Can we interpret the condition number of the basis in geometric terms?
The Geometry of Conditioning

• Consider again the geometric interpretation of condition number of a matrix $A$.

• Roughly speaking, it is the ratio of the largest to smallest axes of the ellipsoid we get by pre-multiplying the points on a unit ball by $A$:

$$\{Ax \mid x \in \mathbb{R}, \|x\| = 1\}$$

• **Question**: What affects the geometry of this ellipsoid?
The Geometry of Conditioning

• Factors affecting the shape of the set \( \{ Ax \mid x \in \mathbb{R}, \|x\| = 1 \} \).
  – The (relative) magnitude of the norms of the rows of \( A \).
  – The “angles” between the rows.

• This is essentially because

\[
|x^\top y| = \|x\| \|y\| \cos \beta
\]

where \( \beta \) is the angle between \( x \) and \( y \).

• Note that condition number is just the “worst case.”

• Using the formula, we can say something about how individual components of the solution to a systems are affected by perturbation.
The Geometry of Conditioning

• Let $r_i$ be the $i^{th}$ row of $A^{-1}$.

• Then it is straightforward to see that if $Ax = b$, we have

$$x_i = r_i^\top b = \|r_i\|\|b\| \cos \beta_i$$

where $\beta_i$ is the angle between $r_i$ and $b$.

• Let $\tilde{x}$ be the solution to $Ax = b + f$ for a given preturbation $f$.

• If $\psi_i$ is the angle between $r_i$ and $f$, then we have

$$\tilde{x}_i = x_i + r_i^\top f = x_i + \|r_i\|\|f\| \cos \psi_i$$

• Further, if $x_i \neq 0$ and $\epsilon_b = \|f\|/\|b\|$, we have

$$\frac{\tilde{x}_i - x_i}{x_i} = \frac{1}{\cos \beta_i} \epsilon_b \cos \psi_i$$

$$= \frac{\|b\|}{\|A\||x|} \frac{\|x\|}{x_i} \frac{\|r_i\|\|A\| \epsilon_b \cos \psi_i}{\|x\|}$$
The Geometry of Conditioning

• The results on the previous slide tell us how to assess the conditioning of the problem of finding individual components of the solution.

• Note that just because a matrix $A$ is ill-conditioned does not mean that the problem of finding each individual component of the solution is ill-conditioned.
  
  – The condition number of the matrix is a worst-case measure over all the component-wise problems.
  
  – There is always one component that exhibits this worst-case behavior.

• The formula on the previous slide tells us that the relative condition of the problem for component $i$ is affected by
  
  – the angle between $r_i$ and $f$
  
  – the angle between $r_i$ and $b$
The Geometry of Conditioning