Advanced Operations Research Techniques IE316

Lecture 6

Dr. Ted Ralphs

Reading for This Lecture

- Bertsimas 3.1-3.2.

Summary of What We've Learned So Far

- We are interested in the extreme points of polyhedra.
- There is a one-to-one correspondence between the extreme points of a polyhedron and the basic feasible solutions.
- We can construct basic solutions by
  - Choosing a basis B of m linearly independent columns of A.
  - Solve the system $Bx_B = b$ to obtain the values of the basic variables.
  - Set $x_N = 0$.
- We can move between adjacent (nondegenerate) basic solutions by removing one column of the basis and replacing it with another.
- In the presence of degeneracy, we might stay at the same extreme point.
- These are the building blocks we need to construct algorithms for solving LPs.

Iterative Search Algorithms

- Many optimization algorithms are *iterative* in nature.
- Geometrically, this means that they move from a given starting point to a new point in a specified *search direction*.
- This search direction is calculated to be both *feasible* and *improving*.
- The process stops when we can no longer find a feasible, improving direction.
- For linear programs, it is always possible to find a feasible improving direction if we are not at an optimal point.
- This is essentially what makes linear programs "easy" to solve.

Feasible and Improving Directions

Definition 1. *Let $\hat{x}$ be an element of a polyhedron $\mathcal{P}$. A vector $d \in \mathbb{R}^n$ is said to be a* feasible direction *if there exists $\theta \in \mathbb{R}_+$ such that $\hat{x} + \theta d \in \mathcal{P}$.*

Definition 2. *Consider a polyhedron $\mathcal{P}$ and the associated linear program $\min_{x \in \mathcal{P}} c^T x$ for $c \in \mathbb{R}^n$. A vector $d \in \mathbb{R}^n$ is said to be an* improving direction *if $c^T d < 0$.*

Notes:

- Once we find a feasible, improving direction, we want to move along that direction as far as possible.
- Recall that we are interested in extreme points.
- The first algorithm we will develop moves between adjacent extreme points using improving directions.

Constructing Feasible Search Directions

- Consider a BFS $\hat{x}$, so that $\hat{x}_N = 0$.
- Any feasible direction must increase the value of at least one of the nonbasic variables (why?).
- We will consider moving in *basic directions* that increase the value of exactly one of the nonbasic variables, say variable j. This means

$$
\begin{aligned}
d_j &= 1 \\
d_i &= 0 \text{ for every nonbasic index } i \neq j
\end{aligned}
$$

- In order to remain feasible, we must also have $Ad = 0$ (why?), which means

$$0 = Ad = \sum_{i=1}^{n} A_i d_i = \sum_{i=1}^{m} A_{B(i)} d_{B(i)} + A_j = B d_B + A_j \Rightarrow d_B = -B^{-1} A_j$$

Constructing Improving Search Directions

- Now we know how to construct feasible search directions—how do we ensure they are improving?
- Recall that we must have $c^T d < 0$.

Definition 3. *Let $\hat{x}$ be a basic solution, let B be an associated basis matrix, and let c_B be the vector of costs of the basic variables. For each j, we define the* reduced cost $\bar{c}_j$ *of variable j by*

$$\bar{c}_j = c_j - c_B^T B^{-1} A_j.$$

- The basic direction associated with variable j is improving if and only if $\bar{c}_j < 0$.
- Note that all basic variables have a reduced cost of 0 (why?).

Optimality Conditions

Theorem 1. *Consider a basic feasible solution $\hat{x}$ associated with a basis matrix B and let $\bar{c}$ be the corresponding vector of reduced costs.*

- *If $\bar{c} \geq 0$, then $\hat{x}$ is optimal.*
- *If $\hat{x}$ is optimal and nondegenerate, then $\bar{c} \geq 0$.*

Notes:

- The condition $\bar{c} \geq 0$ implies there are no feasible improving directions.
- However, $\bar{c}_j < 0$ does not ensure the existence of an improving, feasible direction unless the current BFS is nondegenerate

.

Optimal Bases

Definition 4. *A basis matrix B is said to be optimal if*

- $B^{-1}b \geq 0$, *and*
- $\bar{c} \geq 0$.

Notes:

- An optimal basis always corresponds to an optimal basic feasible solution.
- However, just because the basis is not optimal does not mean the corresponding BFS is not optimal.

An Algorithm for Linear Programming

We will develop the following basic algorithm for linear programming:

1. Find an initial BFS.
2. Compute the reduced costs.
3. Determine an improving feasible direction d.
4. Move as far as possible in direction d to a new BFS.
5. If the new BFS is not optimal, then repeat.

The Step Length

- For now, we will assume that we can find an initial BFS (Step 1).
- We will also assume nondegeneracy.
- We have already seen how to compute the reduced costs and find an improving feasible direction (Steps 2 and 3).
- The distance we move in the computed direction is the *step length*.
- We want to move as far as possible, so the step length is

$$\theta^* = \max\{\theta \geq 0 \mid \hat{x} + \theta d \in \mathcal{P}\}$$

- What really determines the step length?
- Answer: We can keep going until a constraint is violated, i.e., until one of the basic variables becomes zero.

Determining the Step Length

- If $d \geq 0$, then the step length is ∞ and the linear program is unbounded.
- if $d_i < 0$, then $\hat{x}_i + \theta d_i \geq 0 \Rightarrow \theta \leq -\frac{\hat{x}_i}{d_i}$.
- Therefore, we can compute the step length explicitly as

$$\theta^* = \min_{\{i | d_i < 0\}} -\frac{\hat{x}_i}{d_i}$$

- Note that we need only consider the basic variables in this computation.

Determining the Next Solution

- Once we have θ^*, the new feasible solution is $\hat{x} + \theta^* d$. Is this a BFS?
- One variable that was nonbasic now has positive value (the *entering variable*).
- One (at least) variable that was basic now has value 0 (the *leaving variable*).
- If j is the entering variable and i is the leaving variable, define the new set of basic variables by

$$\bar{B}(l) = \begin{cases} B(l), & B(l) \neq i, \\ j, & \text{otherwise.} \end{cases}$$

- Is the corresponding matrix $\bar{B}$ a basis matrix?

Determining the Next Basis

Theorem 2.

- *The columns* $A_{\bar{B}(i)}$ *for* $i \in [1..m]$ *are linearly independent and hence* $\bar{B}$ *does form a basis matrix.*
- *The vector* $\hat{x} + \theta^* d$ *is the BFS corresponding the* $\bar{B}$*.*

Notes:

- This is all we need to ensure the correctness of the algorithm proposed earlier.
- The procedure corresponds to traveling along edges of the polytope between adjacent extreme points until the optimum is reached.

The Simplex Method

A typical iteration of the simplex method:

1. Start with a specified basis matrix B and a corresponding BFS x^0.
2. Compute the reduced cost vector $\bar{c}$. If $\bar{c} \geq 0$, then x^0 is optimal.
3. Otherwise, choose j for which $\bar{c}_j < 0$.
4. Compute $u = B^{-1}A_j$. If $u \leq 0$, then $\theta^* = \infty$ and the LP is unbounded.
5. Otherwise, $\theta^* = \min_{\{i=1,\ldots,m: u_i > 0\}} \frac{x^0_{B(i)}}{u_i}$.
6. Choose l such that $\theta^* = \frac{x^0_{B(l)}}{u_l}$ and form a new basis, replacing $A_{B(l)}$ with A_j. The values of the new basic variables are $x^1_j = \theta^*$ and $x^1_{B(i)} = x^0_{B(i)} - \theta^* u_i$ if $i \neq l$.