CSCI5654 (Linear Programming, Fall 2013)
Lecture-6

Today's Lecture

1. Initialization (overview).
2. Redo the proofs (less handweaving).
3. Degeneracy.
4. Cycling.
5. Ways to avoid cycling.

We solve two problems and discuss initialization Simplex for first 15 minutes.

Initialization

1. Formulate auxilliary problem.
2. Force x_0 to enter and variable with least -ve constant term to leave.
3. Iterate according to basic simplex.
   - Whenever x_0 can leave the basis, force it leave.
4. If optimal value is 0:
   - Remove x_0 from the dictionary.
   - Replace original objective function.
5. If optimal value is < 0, declare solution INFEASIBLE.

Linear Programming Problem

LP in standard form:

$$
\begin{array}{rrcl}
\text{maximize} & \sum_i c_i x_i & & \\
\text{s.t.} & a_{11}x_1 + \cdots + a_{1n}x_n & \leq & b_1 \\
 & a_{21}x_1 + \cdots + a_{2n}x_n & \leq & b_2 \\
 & & \vdots & \\
 & a_{m1}x_1 + \cdots + a_{mn}x_n & \leq & b_m \\
 & x_1, \ldots, x_n & \geq & 0
\end{array}
$$

Auxilliary Problem

Idea: Add new variable x_0.
Modified Problem:

$$
\begin{array}{rrcl}
\text{maximize} & -x_0 & & \\
\text{s.t.} & a_{11}x_1 + \cdots + a_{1n}x_n - x_0 & \leq & b_1 \\
 & a_{21}x_1 + \cdots + a_{2n}x_n - x_0 & \leq & b_2 \\
 & & \vdots & \\
 & a_{m1}x_1 + \cdots + a_{mn}x_n - x_0 & \leq & b_m \\
 & x_0, x_1, \ldots, x_n & \geq & 0
\end{array}
$$

Auxilliary Problem: Theorems

Theorem: The auxilliary LP problem:

1. Is always feasible.
2. Has an optimal solution that is ≤ 0.
3. Optimal solution is 0 iff original LP is feasible.

Proof: Proofs follow from the following observations:

1. Choose arbitrary non-negative values for $x_1, \ldots, x_n$:
   Claim: Can choose a value for x_0 that
   satisfies auxilliary LP constraints.

2. $x_0 \geq 0$ is a constraint and objective is $z = -x_0$. Therefore, $\boxed{z \leq 0}$.

3. $(x_1, \ldots, x_n)$ satisfies the original LP constraints iff $(x_0 = 0, x_1, \ldots, x_n)$ satisfies the auxilliary LP.

Important fact about dictionaries #1

$$
\begin{array}{rcl}
x_{j1} & = & b_1 - \sum_i a_{1i} x_i \\
x_{j2} & = & b_2 - \sum_i a_{2i} x_i \\
 & \vdots & \\
\mathbf{x_{jr}} & = & \mathbf{b_r} - \sum_i \mathbf{a_{ri} x_i} \\
 & \vdots & \\
x_{jm} & = & b_m - \sum_i a_{mi} x_i \\
\hline
z & = & c_0 - \sum c_i x_i
\end{array}
$$

Current Basic Solution: $(x_1 = 0, \ldots, x_n = 0, x_{j_1} = b_1, \ldots, x_{j_m} = b_m)$.

- For entering var. x_e, let $x_e = t$ be the <u>maximum possible increase</u>.
- If we set $x_e = t$ then in the <u>next dictionary</u> D', we have

$$x_{jr} = b_r - a_{re} t\,, \text{ for } r = 1, \ldots, m$$

Fact: Any variable x_{jr} s.t. $b_r - a_{re} t = 0$ is a candidate to leave the basis in D.

Important fact about dictionaries #2

Take dictionary D:

$$\begin{array}{rcl} x_{j1} & = & b_1 - \sum_i a_{1i} x_i \\ x_{j2} & = & b_2 - \sum_i a_{2i} x_i \\ & \vdots & \\ \mathbf{x_{jr}} & = & \mathbf{b_r} - \sum_i \mathbf{a_{ri} x_i} \\ & \vdots & \\ x_{jm} & = & b_m - \sum_i a_{mi} x_i \\ \hline z & = & c_0 - \sum c_i x_i \end{array}$$

Suppose we perform a pivoting step (x_i enters and some x_j leaves). We obtain a new dictionary D'.

D' <u>is algebraically equivalent to</u> D

Auxilliary Dictionary: Final State

If original LP is feasible, initialization phase LP ends with the following dictionary D:

$$\begin{array}{rcll} x_{i_1} & = & \cdots & i_1 \neq 0 \\ & \vdots & & \\ x_{i_m} & = & \cdots & i_m \neq 0 \\ \hline w & = & -x_0 & \end{array}$$

Proof:

- Final dictionary D has optimal value 0.
- Assume (for contradacation's sake) that x_0 is a basic (LHS, dependent) variable in D.
- Let D' be the dictionary previous to D. Cleary $w < 0$ in D' (because it is not yet optimal). But $w = -x_0$. So $x_0 > 0$ in D'.
- Let x_e be the entering variable in D' and x_j be the leaving variable.
- Clearly, $x_0 = 0$ in D. Therefore x_0 was a candidate to leave in the previous dictionary D'.
- Rule for initialization: whenever x_0 was a candidate to leave then it has to leave.
- If x_0 were not independent var. in D, we have clearly violated the rule above.
- Therefore, x_0 must be independent (RHS, non-basic) variable in D.

$\boxed{\text{Degeneracy}}$

Recall: Basic Solution (associated with dictionary)

$$
\begin{array}{rcrrrr}
x_4 & = & 1 & & & -2x_3 \\
x_5 & = & 3 & -2x_1 & +4x_2 & -6x_3 \\
x_6 & = & 2 & +x_1 & -3x_2 & -4x_3 \\
\hline
z & = & & 2x_1 & -x_2 & +8x_3
\end{array}
$$

Q1: What solution is associated with this dictionary?

Q2: x_3 is the entering variable, what variable should leave?

Degeneracy

Definition: A dictionary is degenerate, iff one or more basic variables have the value 0 in the basic solution.

$$\begin{array}{rcrrrr} x_3 & = & .5 & & & -.5x_4 \\ x_5 & = & & -2x_1 & +4x_2 & +3x_4 \\ x_6 & = & & x_1 & -3x_2 & +2x_4 \\ \hline z & = & 4 & +2x_1 & -x_2 & -4x_4 \end{array}$$

Solution: $(x_1 = 0, x_2 = 0, x_4 = 0; x_3 = .5, x_5 = 0, x_6 = 0)$.
Q: If x_1 is the entering variable, what is the leaving variable?

Degeneracy

Prove: If we move from dictionary D to D' and the objective function value does not change then D, D' are degenerate dictionaries.

Proof:

- Let $(x_1, \ldots, x_n)$ be the value of the basic solution associated with D.
- Let x_e be the entering variable we choose.
- Assumed that objective did not increase from $D \to D'$.
- Therefore, increase to x_e is limited to zero by some basic variable x_r (draw a figure).
- Claim that $x_r = 0$ in D and x_i (basic variable in D') will be zero.

Cycling

Cycling: Simplex goes from a dictionary D back to itself.

$$D \to D_1 \to D_2 \cdots D_m \to D\,.$$

Cycling can happen: Page 31,32 of Chvátal.
Prove: Dictionaries $D, D_1, \ldots, D_m$ are all degenerate.

Termination of Simplex

Observation: If a Simplex execution does not terminate,
it must **cycle**. This follows from the following theorem:
Theorem: The number of possible dictionaries for a given problem is <u>finite</u>.
Proof: We will prove that if two dictionaries D, D' have the same set of basic variables $x_1, \ldots, x_m$, then they must be <u>identical</u>.
Given a dictionary with m slack variables and n decision variables, upper bound on number of dictionaries is $\left(\begin{array}{c} n+m \\ m \end{array} \right)$

Avoiding Cycling: Bland's Rule

Rank Variables: $x_1 \prec x_2 \prec \cdots \prec x_{n+m}$. Assume index of variable x_i is i.
Bland's Rule: For each dictionary D:

1. If $x_{e_1}, \ldots, x_{e_k}$ are the possible entering variables, choose the one with the least subscript index.
2. If $x_{r_1}, \ldots, x_{r_j}$ are the possible leaving variables, then choose the one with the least subscript index.

If we choose entering,leaving variables using Bland's rule, then Simplex will not cycle.

Thus far...

- Degeneracy.
- Cycling.
- Termination through Bland's rule.

Complexity

Consider instance with n variables, m constraints.

Upper bound on number of iterations: $\binom{n+m}{m}$.

Q: What does it take to achieve this upper bound?

Klee-Minty Examples

$$
\begin{array}{rcll}
\text{max.} & 10^{n-1}x_1 + 10^{n-2}x_2 + \ldots + 10x_{n-1} + 1x_n & & \\
\text{s.t.} & x_1 & \leq & 1 \\
 & 20x_1 + x_2 & \leq & 100 \\
 & 200x_1 + 20x_2 + x_3 & \leq & 100^2 \\
 & \cdots & & \\
 & 2(\sum_{j=1}^{i-1} 10^{i-j} x_j) + x_i & \leq & 100^{i-1} \\
 & \cdots & & \\
 & x_1, \ldots, x_n & \geq & 0
\end{array}
$$

Theorem: If we follow the largest coefficient rule, then simplex requires $2^n - 1$ iterations to converge.

Average Case Complexity

Result #1: Many random problem instances, generated using various distribution. Expected complexity is polynomial time.
Result #2 [Spielman+Teng, 2001]: If we add random Gaussian noise to the coefficients of a problem, the expected running time becomes polynomial in the size of the initial problem.

$$\begin{array}{rl} \text{max.} & \vec{c}\cdot\vec{x} \\ \text{s.t.} & (A + \underbrace{G(0,\sigma)}_{\text{random matrix}})\vec{x} \leq \vec{b} + \vec{g}(0,\sigma) \end{array}$$

Informally: If we perturb each hard instance enough, we are very likely to find an easy instance of Simplex.

Next Lecture

- Duality
- Complementary Slackness