Carlos Mc Gregor c.mcgregormuro@mail.utoronto.ca

CSC373S: Algorithm Design, Analysis & Complexity

Lecture 23

Monday March 6, 2017
based on notes by Denis Pankratov

Linear Programming (LP) (*cont.*)

Simplex (*cont.*)

(*Midterm cutoff*)

$\max x_1 + 6x_2$
$x_1 \leq 200$
$x_2 \leq 300$
$x_1 + x_2 \leq 400$

$z = x_1 + 6x_2$
$x_3 = 200 - x_1$
$x_4 = 300 - x_2$
$x_5 = 400 - x_1 - x_2$

Variables on the RHS are **NONBASIC** $\{x_1, x_2\}$
Variables on the LHS are **BASIC** $\{x_3, x_4, x_5\}$

Basic Solution:
Set nonbasic variables to 0
$x_1 = 0, x_2 = 0, x_3 = 200, x_4 = 300, x_5 = 400, z = 0$

Rule 1: Choose a *nonbasic* variables that has a positive coefficient in the objective. Such variable is called *entering*. If no such variable, then we are at the **optimal value**, so terminate.

Example: Choose x_2 as entering variable.
Rule 2: Find **leaving** variable
Start increasing value of entering variable *until* some constraint becomes violated. The basic variable corresponding to *tightest* such constraint is called **leaving**.
If no leaving variable, then LP is *unbounded*.

Example: in our example x_4 - *leaving* variable
Rule 3: Exchange entering & leaving variables using the constraint of the leaving variable.
Rewrite the whole LP (exchanges 1 basic variable with 1 nonbasic).
Geometrically, positions neighbor of original feasible solution at *the origin*.

<u>Basic Solution:</u> $x_1 = 0, x_2 = 0, x_3 = 200, x_4 = 300, x_5 = 400, z = 0$
Entering variable x_2
Leaving variable x_4
Pivot: $x_2 = 300 - x_4$

$z = x_1 + 6(300 - x_4)$
$x_3 = 200 - x_1$
$x_2 = 300 - x_4$
$x_5 = 400 - x_1 = (300 - x_4)$

$z = 1800 + x_1 - 6x_4$
$x_3 = 200 - x_1$
$x_2 = 300 - x_4$
$x_5 = 100 - x_1 + x_4$

<u>Basic Solution:</u> $x_1 = 0, x_2 = 300, x_3 = 200, x_4 = 0, x_5 = 100, z = 1800$
Entering variable x_1
Leaving variable x_5
Pivot $x_1 = 100 + x_4 - x_5$

$z = 1900 - x_5 - 5x_4$
$x_3 = 100 - x_4 + x_5$
$x_2 = 300 - x_4$
$x_1 = 100 + x_4 - x_5$

<u>Basic Solution:</u> $x_1 = 100, x_2 = 200, x_3 = 100, x_4 = 0, x_5 = 0, z = 1900$

(*Note:* At different steps, we have different solutions (*i.e.* x_1, x_2))

Step	x_1	x_2
1	0	0
2	0	300
3	100	300

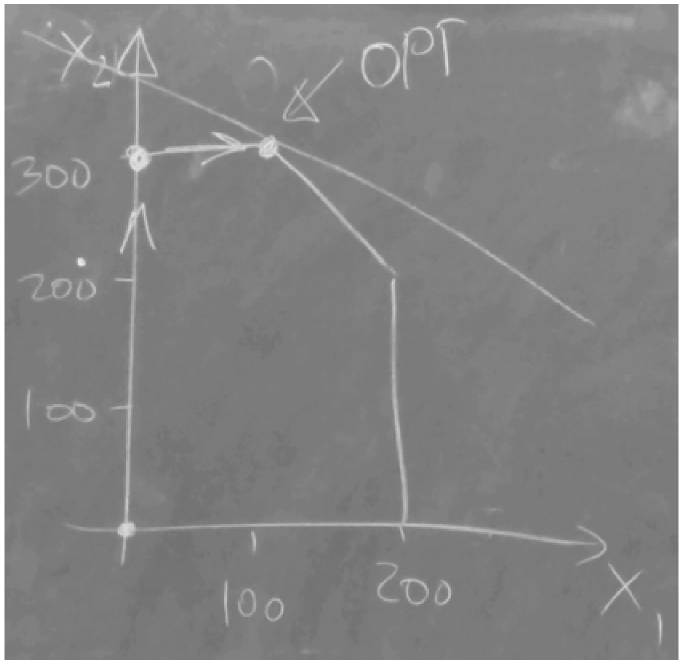

Figure 1: *Visualization*

What if $b \ngeq 0$*?*
0 might not be a feasible starting vertex.
How do we find a feasible vertex?

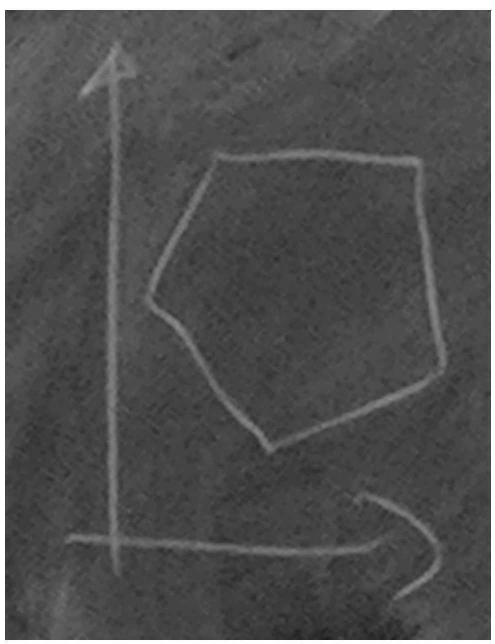

Figure 2: 0 being an *unfeasible* vertex

Reduces to solving LP: Introduce new variable x_0
Original
$\max c^T x$
$a_1^T x \leq b_1$
$\vdots$
$a_m^T x \leq b_m$
$x \geq 0$
$x = (x_1, .., x_n)$

Auxiliary
$\max -x_0$
$a_1^T x - x_0 \leq b_1$
$\vdots$
$a_m^T x - x_0 \leq b_m$
$x_0 \geq 0, x \geq 0$

Theorem: OPT of *auxiliary* is $\varnothing \Leftrightarrow$ original is feasible

Proof: $\Rightarrow$ Let $\bar{x}_0, \bar{x}$ be OPT for *auxiliary*
Since $OPT = 0, \bar{x}_0 = 0$
$\forall j, a_j^T \bar{x} - \bar{x}_0 \leq b_j \Rightarrow a_j^T \bar{x} \leq b_j \Rightarrow \bar{x}$ is feasible with respect to *original*

$\Leftarrow$ Let $\bar{x}$ be a feasible solution of *original.* Then extend it to $\bar{x}_0 = 0$
$\Rightarrow (\bar{x}_0, \bar{x})$ is a feasible solution to *auxiliary*
has value 0 in *auxiliary*

Since OPT of *auxiliary* ≤ 0, then $\bar{x}_0, \bar{x}$ is optimal.
To find a feasible starting point of *original*:

1. set up *auxiliary*
2. starting feasible point for *auxiliary*:

$$x_0 = -min_i b_i, x_1 = x_2 = \cdots = x_n = 0$$

$$\forall j, a_j^T x - x_0 = 0 + \min_i b_i = \min_i b_i \leq b_j$$

3. run *Simplex*
   If OPT of *auxiliary* ¡ 0, then report *original* is infeasible
   Otherwise, use found solution without x_0 as a starting feasible vertex for *original.*
4. Run *Simplex* on *original*

Degeneracy

vertex v is dgenerate if there are several different representations of v as a unique point of intersection of n *hyperplanes*

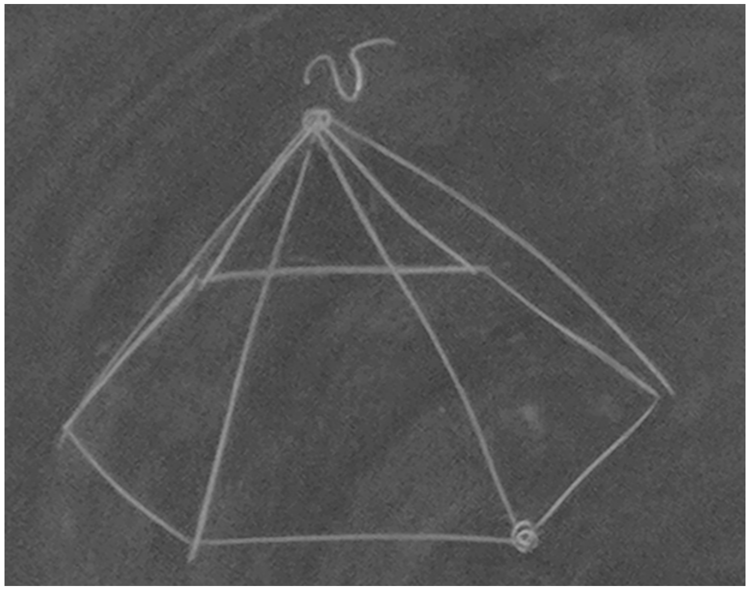

Figure 3: v creates 6 *hyperplanes*, thus making it different from other vertices

Simplex might have to explore these representations w/o improving objective & might get stuck in an *infinite loop* (also known as *cycling*).

Bland's rule avoids cycling: pick entering & leaving variables to be *lexicographically* first ones among all available.

Runtime:
1 pivot operation = $\mathcal{O}(mn)$, where m is the number of rows in $Ax \leq b$, n is $x = (x_1, .., x_n)$
number of pivot operations = $\mathcal{O}\binom{m+n}{n}$, not polynomial

Exercise: estimate $\binom{2n}{n}$ using *Stirling's approximation*