#### CS 473: Algorithms, Fall 2018

## Linear Programming II

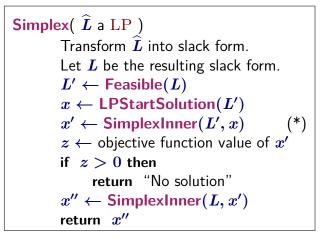
Lecture 19 October 29, 2018

Let  $\mathcal{L}$  be an instance of LP with n variables and m constraints. Then we have the following:

- **1**  $\mathcal{L}$  is always feasible.
- I might not be feasible, but it can be made feasible by changing the value of one of the variables.
- Image constraints and the set of the set
- C might not be feasible, but can be fixed by adding two variable with the correct value (one need two variables because of the equality constraints).
- $\bigcirc$  *L* might not be feasible, and this can not be fixed.

# 19.1: The Simplex Algorithm in Detail

#### Simplex algorithm



#### Simplex algorithm...

- SimplexInner: solves a LP if the trivial solution of assigning zero to all the nonbasic variables is feasible.
- **2** L' = Feasible(L) returns a new LP with feasible solution.
- **(a)** Done by adding new variable  $x_0$  to each equality.
- Set target function in L' to  $\min x_0$ .
- original LP L feasible  $\iff$  LP L' has feasible solution with  $x_0 = 0$ .
- Apply SimplexInner to L' and solution computed (for L') by LPStartSolution(L').
- If  $x_0 = 0$  then have a feasible solution to L.
- **Over Solution in SimplexInner on** *L***.**
- need to describe SimplexInner: solve LP in slack form given a feasible solution (all nonbasic vars assigned value 0).

#### Notations

*B* - Set of indices of basic variables *N* - Set of indices of nonbasic variables n = |N| - number of original variables *b*, *c* - two vectors of constants m = |B| - number of basic variables (i.e., number of inequalities)  $A = \{a_{ij}\}$  - The matrix of coefficients

 $N\cup B=\{1,\ldots,n+m\}$ 

 $m{v}$  - objective function constant.

LP in slack form is specified by a tuple (N, B, A, b, c, v).

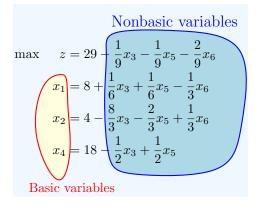
6

#### The corresponding $\operatorname{LP}$

$$egin{aligned} \max & z = v + \sum_{j \in N} c_j x_j, \ & ext{s.t.} & x_i = b_i - \sum_{j \in N} a_{ij} x_j ext{ for } i \in B, \ & x_i \geq 0, \qquad orall i = 1, \dots, n+m. \end{aligned}$$

7

#### Reminder - basic/nonbasic



8

### 19.2: The SimplexInner Algorithm

9

#### The SimplexInner Algorithm

#### Description SimplexInner algorithm:

- LP is in slack form.
- Trivial solution  $x = \tau$  (i.e., all nonbasic variables zero), is feasible.
- objective value for this solution is v.
- **③** Reminder: Objective function is  $z = v + \sum_{j \in N} c_j x_j$ .
- (a)  $x_e$ : nonbasic variable with positive coefficient in objective function.
- Formally: e is one of the indices of  $\left\{ j \mid c_j > 0, j \in N \right\}$ .
- **@**  $x_e$  is the *entering variable* (enters set of basic variables).
- **③** If increase value  $x_e$  (from current value of **0** in au)...
- In one of basic variables is going to vanish (i.e., become zero).

#### Choosing the leaving variable

- $x_e$ : entering variable
- 2  $x_l$ : *leaving* variable vanishing basic variable.
- Increase value of  $x_e$  till  $x_l$  becomes zero.
- How do we now which variable is  $x_l$ ?
- ${igsidentify}$  set all nonbasic to  ${f 0}$  zero, except  $x_e$
- $\ \, {\bf 0} \ \, x_i=b_i-a_{ie}x_e, \ \, {\rm for \ \, all} \ \, i\in B.$
- Require:  $\forall i \in B$   $x_i = b_i a_{ie}x_e \geq 0$ .
- $\textcircled{0} \implies x_e \leq (b_i/a_{ie})$
- $l = \arg\min_i b_i / a_{ie}$
- **(**) If more than one achieves  $\min_i b_i / a_{ie}$ , just pick one.

#### Pivoting on ${\rm x}_{\rm e}...$

- Determined  $x_e$  and  $x_l$ .
- 2 Rewrite equation for  $x_l$  in LP.

 ${\small \textcircled{\sc 0}}$  (Every basic variable has an equation in the LP!)

$$\begin{array}{ll} { \bullet } & x_l = b_l - \sum_{j \in N} a_{lj} x_j \\ \qquad \Longrightarrow \quad x_e = \frac{b_l}{a_{le}} - \sum_{j \in N \cup \{l\}} \frac{a_{lj}}{a_{le}} x_j, \qquad \text{where } a_{ll} = 1. \end{array}$$

- 3 Cleanup: remove all appearances (on right) in LP of  $x_e$ .
- Substituting  $x_e$  into the other equalities, using above.
- Solution Alternatively, do Gaussian elimination remove any appearance of  $x_e$  on right side LP (including objective). Transfer  $x_l$  on the left side, to the right side.

#### Pivoting continued...

- **1** End of this process: have new *equivalent* LP.
- **2** basic variables:  $B' = (B \setminus \{l\}) \cup \{e\}$
- non-basic variables:  $N' = (N \setminus \{e\}) \cup \{l\}.$
- End of this *pivoting* stage: LP objective function value increased.
- Made progress.
- LP is completely defined by which variables are basic, and which are non-basic.
- Pivoting never returns to a combination (of basic/non-basic variable) already visited.
- ...because improve objective in each pivoting step.
- Can do at most  $\binom{n+m}{n} \leq \left(\frac{n+m}{n} \cdot e\right)^n$ .
- 0 examples where  $2^n$  pivoting steps are needed.

#### Simplex algorithm summary...

- Each pivoting step takes polynomial time in n and m.
- **2** Running time of **Simplex** is exponential in the worst case.

14

In practice, **Simplex** is extremely fast.

# Pivoting with zeroes?

Consider a pivoting step, with  $x_e$  as the entering variable, and  $x_{\ell}$  as the leaving variable, with the relevant constraint in the LP being:

$$x_{\ell} = 0 - \sum_{j \in N} a_{lj} x_j.$$

- Doing the pivoting step would involve division by zero, and as such the Simplex algorithm would fail.
- 2 There is no problem.
- In an LP the constant in a constraint can never be zero, so this is an impossible scenario.
- If there is any problem, it can be solved by choosing a different entering/leaving variables.
- The pivoting step would not improve the LP objective function.
  Simplex might pivot in a loop forever.

#### Degeneracies

- **Simplex** might get stuck if one of the  $b_i$ s is zero.
- More than > m hyperplanes (i.e., equalities) passes through the same point.
- Result: might not be able to make any progress at all in a pivoting step.
- Solution I: add tiny random noise to each coefficient.
  Can be done symbolically.
  Intuitively, the degeneracy, being a local phenomena on the polytope disappears with high probability.

#### Degeneracies – cycling

- Might get into cycling: a sequence of pivoting operations that do not improve the objective function, and the bases you get are cyclic (i.e., infinite loop).
- Solution II: *Bland's rule*.
  Always choose the lowest index variable for entering and leaving out of the possible candidates.
  (Not prove why this work but it does.)

### $19.2.1: \ \ {\rm Correctness} \ {\rm of} \ {\rm linear} \ {\rm programming}$

#### Correctness of $\operatorname{LP}$

#### Definition

A solution to an LP is a *basic solution* if it the result of setting all the nonbasic variables to zero.

Simplex algorithm deals only with basic solutions.

#### Theorem

For an arbitrary linear program, the following statements are true:

- If there is no optimal solution, the problem is either infeasible or unbounded.
- **2** If a feasible solution exists, then a basic feasible solution exists.
- **③** If an optimal solution exists, then a basic optimal solution exists.

Proof: is constructive by running the simplex algorithm.

# $\begin{array}{c} 19.2.2\text{:} \quad \text{On the ellipsoid method and interior point} \\ \text{methods} \end{array}$

#### On the ellipsoid method and interior point methods

- **Simplex** has exponential running time in the worst case.
- ellipsoid method is weakly polynomial.
  It is polynomial in the number of bits of the input.
- Skhachian in 1979 came up with it. Useless in practice.
- In 1984, Karmakar came up with a different method, called the interior-point method.
- Solution Also weakly polynomial. Quite useful in practice.
- Result in arm race between the interior-point method and the simplex method.
- BIG OPEN QUESTION: Is there *strongly* polynomial time algorithm for linear programming?

# Solving LPs without ever getting into a loop - symbolic perturbations

Details in the class notes.















