Chapter 19

Linear Programming II

NEW CS 473: Theory II, Fall 2015
October 29, 2015

19.1 The Simplex Algorithm in Detail

19.1.0.1 Simplex algorithm

```
Simplex( L̂ a LP )
        Transform L̂ into slack form.
        Let L be the resulting slack form.
        L' ← Feasible(L)
        x ← LPStartSolution(L')
        x' ← SimplexInner(L', x)          (*)
        z ← objective function value of x'
        if  z > 0 then
             return  "No solution"
        x'' ← SimplexInner(L, x')
        return  x''
```

19.1.0.2 Simplex algorithm...

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

19.1.0.3 Notations

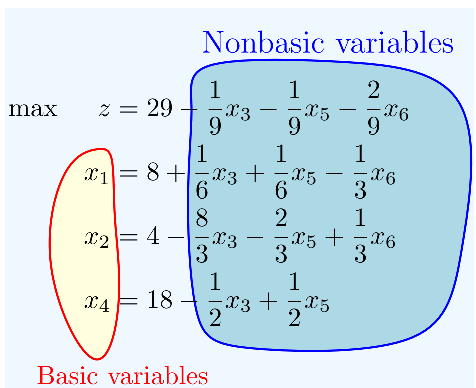
B - Set of indices of basic variables
N - Set of indices of nonbasic variables
$n = |N|$ - number of original variables
b, c - two vectors of constants
$m = |B|$ - number of basic variables (i.e., number of inequalities)
$A = \{a_{ij}\}$ - The matrix of coefficients
$N \cup B = \{1, \dots, n+m\}$
v - objective function constant.

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

19.1.0.4 The corresponding LP

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

19.1.0.5 Reminder - basic/nonbasic



19.2 The SimplexInner Algorithm

19.2.0.1 The SimplexInner Algorithm

Description **SimplexInner** algorithm:
(A) LP is in slack form.
(B) Trivial solution $x = \tau$ (i.e., all nonbasic variables zero), is feasible.
(C) objective value for this solution is v.
(D) Reminder: Objective function is $z = v + \sum_{j \in N} c_j x_j$.
(E) x_e: nonbasic variable with positive coefficient in objective function.
(F) Formally: e is one of the indices of $\left\{ j \;\middle|\; c_j > 0, j \in N \right\}$.
(G) x_e is the ***entering variable*** (enters set of basic variables).
(H) If increase value x_e (from current value of 0 in τ)...
(I) ... one of basic variables is going to vanish (i.e., become zero).

19.2.0.2 Choosing the leaving variable

(A) x_e: ***entering variable***

(B) x_l: ***leaving*** variable – vanishing basic variable.
(C) increase value of x_e till x_l becomes zero.
(D) How do we now which variable is x_l?
(E) set all nonbasic to 0 zero, except x_e
(F) $x_i = b_i - a_{ie}x_e$, for all $i \in B$.
(G) Require: $\forall i \in B \quad x_i = b_i - a_{ie}x_e \geq 0$.
(H) $\implies x_e \leq (b_i/a_{ie})$
(I) $l = \arg\min_i b_i/a_{ie}$
(J) If more than one achieves $\min_i b_i/a_{ie}$, just pick one.

19.2.0.3 Pivoting on x_e...

(A) Determined x_e and x_l.
(B) Rewrite equation for x_l in LP.
 (A) (Every basic variable has an equation in the LP!)
 (B) $x_l = b_l - \sum_{j \in N} a_{lj}x_j$

$$\implies \qquad x_e = \frac{b_l}{a_{le}} - \sum_{j \in N \cup \{l\}} \frac{a_{lj}}{a_{le}} x_j, \qquad \text{where } a_{ll} = 1.$$

(C) Cleanup: remove all appearances (on right) in LP of x_e.
(D) Substituting x_e into the other equalities, using above.
(E) 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.

19.2.0.4 Pivoting continued...

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

19.2.0.5 Simplex algorithm summary...

(A) Each pivoting step takes polynomial time in n and m.
(B) Running time of **Simplex** is exponential in the worst case.
(C) In practice, **Simplex** is extremely fast.

19.2.0.6 Degeneracies

(A) **Simplex** might get stuck if one of the b_is is zero.
(B) More than $> m$ hyperplanes (i.e., equalities) passes through the same point.
(C) Result: might not be able to make any progress at all in a pivoting step.

(D) 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.

19.2.0.7 Degeneracies – cycling

(A) 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).
(B) 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 Correctness of linear programming

19.2.1.1 Correctness of LP

Definition 19.2.1. 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 19.2.2. *For an arbitrary linear program, the following statements are true:*
(A) If there is no optimal solution, the problem is either infeasible or unbounded.
(B) If a feasible solution exists, then a basic feasible solution exists.
(C) If an optimal solution exists, then a basic optimal solution exists.

Proof: is constructive by running the simplex algorithm.

19.2.2 On the ellipsoid method and interior point methods

19.2.2.1 On the ellipsoid method and interior point methods

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

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

Details in the class notes.