The simplex algorithm: termination

Theorem

Assume that the simplex method is applied to an LP, and that each generated basic feasible solution is nondegenerate ($x_B > 0$ at each iteration). Then, in a finite number of iterations, the method finds an optimal solution or determines that the problem is unbounded.

For proof, see Theorem 5.1 in book.

- Performance in practice: very good
- Requires typically between m and $3m$ iteration
  - In general much less than the total number of possible basic solutions

The simplex algorithm: termination

Ex.:

- $n = 150$ variables, $m = 50$ constraints (a small problem!)
- Number of basic solutions, upper bound: $\binom{n}{m} \approx 10^{40}$ (many may be infeasible though)
- Assume: algorithms requires 1 ns per basis
- Takes 10^{23} years to examine all bases!
- $3m$ iterations takes 150 ns

Artificial LPs can be constructed that forces the simplex algorithm to examine every possible basis

This behavior have not been observed for “real” problems

The simplex method: initialization

Simplex method starts with a basic feasible solution (BFS). How to get this?

Easy for problems in inequality form!

$$\begin{aligned} Ax &\le b \\ x &\ge 0 \end{aligned} \quad \text{slacks} \Rightarrow \quad \begin{aligned} Ax + s &= b \\ x, s &\ge 0 \end{aligned}$$

$$\begin{pmatrix} A & I \end{pmatrix} \begin{pmatrix} x \\ s \end{pmatrix} = b$$

$$\begin{pmatrix} x \\ s \end{pmatrix} \ge 0$$

$x_B = s$, $x_N = x = 0$ is a BFS if $b \ge 0$

For other problems, not so easy

The “two-phase” method

Consider

$$\begin{aligned} \min c^{\mathsf{T}} x \ &\text{subject to} \\ Ax &= b \\ x &\ge 0 \end{aligned}$$

with $b \ge 0$ (can always be obtained by multiplying constraint rows with -1 if needed)

Phase I: Solve an *auxiliary* LP to obtain an initial basic feasible solution
Phase II: Solve the original LP

For phase I, introduce the *artificial variables* y and consider

$$\begin{aligned} \min \sum_{i=1}^{m} y_i \text{ subject to} & \\ Ax + y = b & \\ x, y \geq 0 & \end{aligned} \tag{AP}$$

an LP in x, y. Solve with simplex method. An initial BFS is $x = 0$, $y = b$ (since $b \geq 0$).

If there is a feasible solution to $Ax = b$, $x \geq 0$, (AP) has a solution with $\min \sum_{i=1}^{m} y_i = 0$. The simplex solution will be basic with respect to A.

Solving (AP) to get an initial BFS usually works. Possible problems:

(a) If $\min \sum_{i=1}^{m} y_i > 0$, then $Ax = b$, $x \geq 0$ has no feasible solution!

(b) If $\min \sum_{i=1}^{m} y_i = 0$ and final solution of (AP) is degenerate, might happen that some $y_i = 0$ is in final basis. These basic variables can be exchanged with non-basic x_i variables (§ 5.5.1 in book)

Simplex algorithm: computational issues

LP problems often large. Important with efficient implementation of the steps in algorithm.

Computationally expensive parts:

- $x_B = B^{-1}b$, i. e. solving $Bx_B = b$ (generate basic solutions)
- $y = B^{-\mathsf{T}}c_B$, i. e. solving $B^{\mathsf{T}}y = c_B$ (generate simplex multipliers)
- $\hat{c}_N^{\mathsf{T}} = c_N^{\mathsf{T}} - y^{\mathsf{T}}N$ ("pricing": generate reduced costs). Note: N has typically **many** columns
- $\hat{a}_p = B^{-1}a_p$, i. e. solving $B\hat{a}_p = a_p$ (to generate step)

Solving linear system:

- LU factorization $B = LU$ (takes $O(m^3)$ floating point operations)
- Solve LUx_B it two steps:
  - First $L\tilde{x}_B = b$ $(O(m^2)$ flops)
  - Then $Ux_B = \tilde{x}_B$ $(O(m^2)$ flops)

But new $\tilde{B}$ differs from old B only in one column

Possible to **update** $U \to \tilde{U}$ and form a new "L-like" factor so that $\tilde{B} = \tilde{L}\tilde{U}$ to much lower cost than $O(m^3)$

Pricing: compute $y^{\mathsf{T}}N$ element by element and stop when the first negative component of $c_N^{\mathsf{T}} - y^{\mathsf{T}}N$ is encountered