Running Time of Simplex Algorithm

CSCI 4113/6101

Cycling

Pivoting:

- A non-basic variable z_j with $c_j > 0$ enters basis.
- A basic variable z_{j_k} with $k = \operatorname{argmin} \{ \frac{b_i}{a_{ij}} \mid a_{ij} > 0 \}$ leaves basis.

Cycling

Pivoting:

- A non-basic variable z_j with $c_j > 0$ enters basis.
- A basic variable z_{j_k} with $k = \operatorname{argmin} \left\{ \frac{b_i}{a_{ij}} \mid a_{ij} > 0 \right\}$ leaves basis.

If $b_k = 0$, then z_j remains $0 \Rightarrow$ no increase in objective function value.

Cycling

Pivoting:

- A non-basic variable z_j with $c_j > 0$ enters basis.
- A basic variable z_{j_k} with $k = \operatorname{argmin} \left\{ \frac{b_i}{a_{ij}} \mid a_{ij} > 0 \right\}$ leaves basis.

If $b_k = 0$, then z_j remains 0 $\Rightarrow$ no increase in objective function value.

Worse: The next iteration may move z_j out of the basis an z_{j_k} back in.

Cycling

Pivoting:

- A non-basic variable z_j with $c_j > 0$ enters basis.
- A basic variable z_{j_k} with $k = \operatorname{argmin} \left\{ \frac{b_i}{a_{ij}} \mid a_{ij} > 0 \right\}$ leaves basis.

If $b_k = 0$, then z_j remains 0 $\Rightarrow$ no increase in objective function value.

Worse: The next iteration may move z_j out of the basis an z_{j_k} back in. $\Rightarrow$ Simplex may cycle through the same solutions forever.

Bland's Anti-Cycling Rule

- Of all non-basic variables that can enter the basis ($c_j > 0$), choose the one with smallest index j.

Bland's Anti-Cycling Rule

- Of all non-basic variables that can enter the basis ($c_j > 0$), choose the one with smallest index j.
- Of all basic variables that can leave the basis ($k = \text{argmin} \left\{ \frac{b_i}{a_{ij}} \mid a_{ij} > 0 \right\}$), choose the one with smallest index k.

Bland's Anti-Cycling Rule

- Of all non-basic variables that can enter the basis ($c_j > 0$), choose the one with smallest index j.
- Of all basic variables that can leave the basis ($k = \text{argmin} \{ \frac{b_i}{a_{ij}} \mid a_{ij} > 0 \}$), choose the one with smallest index k.

Lemma: Bland's rule guarantees that there are no two iterations of the Simplex Algorithm whose LPs have the same basis.

Running Time of Simplex Algorithm

m = # constraints / basic variables

n = # non-basic variables

Theorem: When using Bland's rule, the Simplex Algorithm runs for at most $\binom{m+n}{n}$ iterations.