What happened at the degenerate vertex?

From last lecture, we executed one iteration of simplex starting at a degenerate vertex and got a step of $\alpha = 0$. What happened?

At degenerate vertex $x_0$, the active set is $\{1, 2, 5\}$ with active-set matrix

$$A_a = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 2 \end{pmatrix}.$$ 

We chose $W_0 = \{5, 2\}$ and computed the direction $p_0$.

However,

$$A_a p_0 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} \not\geq 0$$

so $p_0$ is not a feasible direction!
Implications of Degeneracy
At a degenerate vertex, the simplex method will keep trying to find a feasible direction using a nonsingular subset of the rows in $A_a$.

During this process, $x_k$ remains “stalled” at the same value until a feasible descent direction is found and we “break free” of the vertex.

This phenomenon is known as “stalling”.

If the simplex method stalls at $x_k$, then

$$x_k = x_{k+1} = \cdots = x_{k+j} \neq x_{k+j+1}$$

Recall that if the vertex $x_k$ is nondegenerate then:

$$W_k = A(x_k) \text{ (the active set)} \quad \text{and} \quad A_k = A_a(x_k)$$

**Theorem**

*If every vertex is nondegenerate, then the simplex method will either declare the LP to be unbounded or find $x^*$ in a finite number of iterations.*
Proof: Under the nondegeneracy assumption, we show that
\[ c^T x_0 > c^T x_1 > c^T x_2 > \cdots > c^T x_k > \cdots \]

\[ \Rightarrow \text{we cannot return to a vertex with a larger objective} \]
\[ \Rightarrow \text{every vertex is visited at most once} \]

The number of vertices is finite.
\[ \Rightarrow \text{for some vertex } x_\ell \text{ it must hold that } c^T x_\ell = c^T x^* \]

First, we show that \( c^T x_k > c^T x_{k+1} \).
\[ x_{k+1} = x_k + \alpha_k p_k \implies c^T x_{k+1} = c^T x_k + \alpha_k c^T p_k \]

By construction, \( c^T p_k < 0 \), so it remains to show that \( \alpha_k > 0 \).
\[ \alpha_k = \sigma_t = \frac{r_t(x_k)}{-a_t^T p_k} \text{ with } a_t^T p_k < 0 \]

\[ \Rightarrow \alpha_k > 0 \text{ if } r_t(x_k) > 0. \]
\[ \Rightarrow \alpha_k > 0 \text{ if the blocking constraint is inactive at } x_k. \]
It holds that \( a_j^T p_k \geq 0 \) for every constraint in the working set.

But \( a_t^T p_k < 0 \)

\[ \Rightarrow t \notin W_k (= A(x_k)), \text{ i.e., the blocking constraint is inactive.} \]

\[ \Rightarrow r_k(x_k) \neq 0 \]

\[ \Rightarrow \alpha_k > 0 \]

\[ \Rightarrow c^T x_k > c^T x_{k+1}. \]

The effects of degeneracy

The simplex method may \textit{stall} at a degenerate vertex \( x_k \).

Nevertheless, it will \textit{usually} find a feasible direction eventually.
BUT... if, at some iteration $k + j$, we have

$$x_k = x_{k+1} = \cdots = x_{k+j} \quad \text{and} \quad W_{k+j} = W_k$$

then we are doomed to repeat the same steps

again, and again, and again, and again, and again, and again ...

This phenomenon is known as “cycling”.

If the simplex method cycles, it never terminates.

Why does stalling and cycling occur?

... because there are many choices for $w_s$ and $t$, the indices of the constraints that enter or leave the working set.

The method cycles when ties are broken with the same choices.
Example:

\[
\begin{align*}
\text{minimize} & \quad -2.3x_1 - 2.15x_2 + 13.55x_3 + 0.4x_4 \\
\text{subject to} & \quad -0.4x_1 - 0.20x_2 + 1.40x_3 + 0.2x_4 \geq 0 \\
& \quad 7.8x_1 + 1.40x_2 - 7.80x_3 - 0.4x_4 \geq 0 \\
& \quad x_i \geq 0, \quad i = 1, 2, 3, 4,
\end{align*}
\]

This LP is unbounded, but has a degenerate vertex at \( x = 0 \).

i.e., an "unbounded solution" indication means that we have broken free of the degenerate vertex.

See Julia example...
Degeneracy *almost always* implies stalling

Stalling *very rarely* implies cycling

Cycling is very unusual in practice.

Why is this?

---

**In theory**, the constraint residual has more than $n$ zero elements

$$r_a(x_k) = A_a x_k - b_a = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad m_a > n$$

But **in practice**, computer calculations are not exact.

$$r_a(x_k) = A_a x_k - b_a = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_{m_a} \end{pmatrix}$$
In theory . . .

In practice . . .
Why does stalling and cycling occur?

...because there are many choices for $w_s$ and $t$, the indices of the constraints that enter or leave the working set.

Cycling can be avoided by choosing $w_s$ and $t$ with a rule that never allows repeats.

Bland’s Least Index Rule
1. Choosing $w_s$

Choose the negative multiplier associated with the least row index in $A$.

$$w_s = \min \{ w_j : \text{multiplier } j \text{ is negative} \}$$

Example:

$$\mathcal{W}_k = \{ 1, 9, 6, 2, 4, 10 \}$$

$$\lambda_k = \{ 120.0, -10.0, 8.0, -2.4, 6.0, -1.0 \}$$

\[ \text{Dantzig least index} \]

2. Choosing $t$

Choose the blocking constraint with the least row index in $A$.

$$\sigma_i = \begin{cases} \frac{r(x_k)}{-a_i^T p_k} & \text{if } a_i^T p_k < 0 \\ +\infty & \text{otherwise} \end{cases}$$

Define $\alpha_k = \min \{ \sigma_j \}$

Set $t = \min \{ j : \sigma_j = \alpha_k \}$
Result (Convergence with Bland’s rule)
Consider an LP for which an initial vertex is known. Then the simplex method, using Bland’s rule to choose the constraints to leave and enter the working set, is guaranteed to terminate.

- If the optimal objective is bounded, the algorithm terminates at an optimal vertex.
- If the objective value is unbounded below in the feasible region, the algorithm terminates with an “unbounded” indication.

**Proof:** See Section 4.4.3 of the Class Text.

Bland’s rule is not useful computationally.

- it usually needs more iterations than the Dantzig rule.

Better anti-cycling rules are based on constraint perturbation.

Nevertheless, Bland’s rule is useful as a theoretical tool.
Result (Carathéodory’s theorem)

A vertex \( x_0 \) with active-set matrix \( A_a \) is a solution of an LP if and only if there is a nonnegative basic solution of \( A_a^T \lambda_a = c \).

Proof: If \( x_0 \) is a nondegenerate vertex then \( A_a \) is nonsingular, \( \lambda_a \) is unique and the result follows.

Assume that \( x_0 \) is a degenerate vertex, with \( A_a^T = \boxed{ } \)

Consider the auxiliary problem:

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad A_a x \geq A_a x_0 \\
\end{align*}
\]

\( x_0 \) is an initial vertex for this problem.

Starting at \( x_0 \), run the simplex method with Bland’s rule.

The simplex method solves a sequence of square systems

\[
A_w^T \lambda_w = c \\
A_w p = e_s
\]

and must either terminate or declare the problem unbounded.
EITHER (A) \( x_0 \) is optimal

OR (B) there is an unbounded direction \( p \)

In other words:

EITHER (A) \( x_0 \) is optimal \( \Rightarrow c = A_w^T \lambda_w \) for some \( \lambda_w \geq 0 \)

OR (B) \( \exists p \) such that \( c^T p < 0 \), \( A_w p = e_s \) and \( A_p \geq 0 \)

("\( \exists p \)" means "there exists a \( p \")

From the preceding slide:

EITHER (A) \( x_0 \) is optimal \( \Rightarrow c = A_w^T \lambda_w \) for some \( \lambda_w \geq 0 \)

OR (B) \( \exists p \) such that \( c^T p < 0 \), \( A_w p = e_s \) and \( A_p \geq 0 \)

(B) \( \Rightarrow \) that there exists a feasible descent direction for the original LP.

(A) \( \Rightarrow c = A_w^T \lambda_w \) for some \( \lambda_w \geq 0 \).

\( \Rightarrow A_w^T \) is a column basis for the equations \( A_s^T \lambda = c \).

The "scattered" \( \lambda_w \) is a nonnegative basic solution of \( A_s^T \lambda = c \).