Agenda

We’ve done
- Greedy Method
- Divide and Conquer
- Dynamic Programming
- Network Flows & Applications
- NP-completeness

Now
- Linear Programming and the Simplex Method

Linear Programming Motivation: The Diet Problem

Setting
- $n$ foods (beef, apple, potato chips, pho, bún bò, etc.)
- $m$ nutritional elements (vitamins, calories, etc.)
- each gram of $j$th food contains $a_{ij}$ units of nutritional element $i$
- a good meal needs $b_i$ units of nutritional element $i$
- each gram of $j$th food costs $c_j$

Objective
- design the most economical meal yet dietarily sufficient
- (Halliburton must solve this problem!)
The Diet Problem as a Linear Program

Let \( x_j \) be the weight of food \( j \) in a dietarily sufficient meal.

\[
\begin{align*}
\text{min} & \quad c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\
\text{subject to} & \quad a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n \geq b_1 \\
& \quad a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n \geq b_2 \\
& \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
& \quad a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n \geq b_m \\
& \quad x_j \geq 0, \forall j = 1, \ldots, n,
\end{align*}
\]

Linear Programming Motivation: The Max-Flow Problem

Maximize the value of \( f \):

\[
\text{val}(f) = \sum_{e=(s,v) \in E} f_e
\]

Subject to

\[
\begin{align*}
0 & \leq f_e \leq c_e, \quad \forall e \in E \\
\sum_{e=(u,v) \in E} f_e - \sum_{e=(v,w) \in E} f_e & = 0, \quad \forall v \neq s, t
\end{align*}
\]
Formalizing the Linear Programming Problem

Linear objective function

\[
\max \text{ or } \min -\frac{8}{3}x_1 + 2x_2 + x_3 - 6x_4 + x_5
\]

Linear constraints, can take many forms

- **Inequality constraints**
  \[
  3x_1 + 4x_5 - 2x_6 \geq 3 \\
  2x_1 + 2x_2 + x_3 \leq 0
  \]

- **Equality constraints**
  \[-x_2 - x_4 + x_3 = -3\]

- **Non-negativity constraints** (special case of inequality)
  \[x_1, x_5, x_7 \geq 0\]

Some notational conventions

All vectors are column vectors

\[
c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}
\]
Linear Program: Standard Form

\[
\begin{align*}
\min / \max & \quad c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\
\text{subject to} & \quad a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n = b_1 \\
& \quad a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n = b_2 \\
& \quad \vdots \\
& \quad a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n = b_m \\
& \quad x_j \geq 0, \forall j = 1, \ldots, n,
\end{align*}
\]

or, in matrix notations,

\[
\min / \max \{ c^T x \mid Ax = b, x \geq 0 \}
\]

Linear Program: Canonical Form – min Version

\[
\begin{align*}
\min & \quad c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\
\text{subject to} & \quad a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n \geq b_1 \\
& \quad a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n \geq b_2 \\
& \quad \vdots \\
& \quad a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n \geq b_m \\
& \quad x_j \geq 0, \forall j = 1, \ldots, n,
\end{align*}
\]

or, in matrix notations,

\[
\min \{ c^T x \mid Ax \geq b, x \geq 0 \}
\]
Linear Program: Canonical Form – max Version

\[
\begin{align*}
\text{max } & \quad c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\
\text{subject to } & \quad a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n \leq b_1 \\
& \quad a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n \leq b_2 \\
& \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \leq \quad \vdots \\
& \quad a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n \leq b_m \\
& \quad x_j \geq 0, \forall j = 1, \ldots, n,
\end{align*}
\]

or, in matrix notations,

\[
\max \left\{ c^T x \mid Ax \leq b, x \geq 0 \right\}
\]

Conversions Between Forms of Linear Programs

- \( \max c^T x = \min (-c)^T x \)
- \( \sum_j a_{ij} x_j = b_i \) is equivalent to \( \sum_j a_{ij} x_j \leq b_i \) and \( \sum_j a_{ij} x_j \geq b_i \).
- \( \sum_j a_{ij} x_j \leq b_i \) is equivalent to \( -\sum_j a_{ij} x_j \geq -b_i \)
- \( \sum_j a_{ij} x_j \leq b_i \) is equivalent to \( \sum_j a_{ij} x_j + s_i = b_i, s_i \geq 0 \). The variable \( s_i \) is called a \textit{slack variable}.
- When \( x_j \leq 0 \), replace all occurrences of \( x_j \) by \( -x'_j \), and replace \( x_j \leq 0 \) by \( x'_j \geq 0 \).
- When \( x_j \) is not restricted in sign, replace it by \( (u_j - v_j) \), and \( u_j, v_j \geq 0 \).
Write
\[
\begin{align*}
\text{min} & \quad x_1 - x_2 + 4x_3 \\
\text{subject to} & \quad 3x_1 - x_2 - x_2 + 2x_4 \geq 4 \\
& \quad x_1 + x_3 \leq -3 \\
& \quad x_1, x_2 \geq 0
\end{align*}
\]
in standard (min / max) form and canonical (min / max) form.

LP Geometry: Example 1

\[
\begin{align*}
\text{max} & \quad 2x + y \\
\text{subject to} & \quad -2x + y \leq 2 \\
& \quad 5x + 3y \leq 15 \\
& \quad x + y \leq 4 \\
& \quad x \geq 0, y \geq 0
\end{align*}
\]
LP Geometry: Example 2

\[
\begin{align*}
\text{max} & \quad 2x + y \\
\text{subject to} & \quad 2x + 3y \geq 8 \\
& \quad 8x + 3y \geq 12 \\
& \quad 4x + 3y \geq 24
\end{align*}
\]

Example 2 – Feasible Region
Example 2 – Objective Function

Half Space and Hyperplane

Each inequality $a^T x \geq b$ defines a half-space.

Each equality $a^T x = b$ defines a hyperplane.
A polyhedron is the set of points $x$ satisfying $Ax \leq b$ (or equivalently $A'x \geq b'$)

$\begin{align*}
3x_1 + 2x_2 + 4x_3 & \geq 9 \\
x_1 + x_2 + 2x_3 & \leq 4 \\
2x_1 + 3x_3 & \leq 5 \\
4x_1 + x_2 + 3x_3 & \leq 7 \\
x_1, x_2, x_3 & \geq 0.
\end{align*}$

Someone claims $x^T = [1 \ 3 \ 0]$ is optimal with cost 9.

Feasibility is easy. **How could we verify optimality?**

We ask the person for a proof!
Her Proof of the Optimality of $x$

\[
\frac{5}{3} \cdot (x_1 + x_2 + 2x_3) \leq \frac{5}{3} \cdot 4
\]
\[
0 \cdot (2x_1 + 3x_3) \leq 0 \cdot 5
\]
\[
\frac{1}{3} \cdot (4x_1 + x_2 + 3x_3) \leq \frac{1}{3} \cdot 7
\]

\[
\Rightarrow 3x_1 + 2x_2 + 4x_3 \leq 9
\]

Done!

**OK, How Did ... I do that?**

Beside listening to Numa Numa, find non-negative multipliers

\[
y_1 \cdot (x_1 + x_2 + 2x_3) \leq 4y_1
\]
\[
y_2 \cdot (2x_1 + 3x_3) \leq 5y_2
\]
\[
y_3 \cdot (4x_1 + x_2 + 3x_3) \leq 7y_3
\]

\[
\Rightarrow (y_1 + 2y_2 + 4y_3)x_1 + (y_1 + y_3)x_2 + (2y_1 + 3y_2 + 3y_3)x_3 \leq (4y_1 + 5y_2 + 7y_3)
\]

Want LHS to be like $3x_1 + 2x_2 + 4x_3$. Thus, as long as

\[
y_1 + 2y_2 + 4y_3 \geq 3
\]
\[
y_1 + y_3 \geq 2
\]
\[
2y_1 + 3y_2 + 3y_3 \geq 4
\]
\[
y_1, y_2, y_3 \geq 0.
\]

we have

\[
3x_1 + 2x_2 + 4x_3 \leq 4y_1 + 5y_2 + 7y_3
\]
How to Get the Best Multipliers

Answer: minimize the upper bound.

\[
\begin{align*}
\text{min} & \quad 4y_1 + 5y_2 + 7y_3 \\
y_1 & + 2y_2 + 4y_3 \geq 3 \\
y_1 & + y_3 \geq 2 \\
2y_1 & + 3y_2 + 3y_3 \geq 4 \\
y_1, y_2, y_3 & \geq 0.
\end{align*}
\]

What We Have Just Shown

If \(x\) is feasible for the Primal Program

\[
\begin{align*}
\text{max} & \quad 3x_1 + 2x_2 + 4x_3 \\
\text{subject to} & \quad x_1 + x_2 + 2x_3 \leq 4 \\
& \quad 2x_1 + 3x_3 \leq 5 \\
& \quad 4x_1 + x_2 + 3x_3 \leq 7 \\
x_1, x_2, x_3 & \geq 0.
\end{align*}
\]

and \(y\) is feasible for the Dual Program

\[
\begin{align*}
\text{min} & \quad 4y_1 + 5y_2 + 7y_3 \\
y_1 & + 2y_2 + 4y_3 \geq 3 \\
y_1 & + y_3 \geq 2 \\
2y_1 & + 3y_2 + 3y_3 \geq 4 \\
y_1, y_2, y_3 & \geq 0.
\end{align*}
\]

then

\[
3x_1 + 2x_2 + 4x_3 \leq 4y_1 + 5y_2 + 7y_3
\]
Primal-Dual Pairs - Canonical Form

\[ \begin{align*}
\text{min} & \quad c^T x \quad \text{(primal/dual program)} \\
\text{subject to} & \quad Ax \geq b \\
& \quad x \geq 0
\end{align*} \]

\[ \begin{align*}
\text{max} & \quad b^T y \quad \text{(dual/primal program)} \\
\text{subject to} & \quad A^T y \leq c \\
& \quad y \geq 0.
\end{align*} \]

Note
The dual of the dual is the primal!

Primal-Dual Pairs - Standard Form

\[ \begin{align*}
\text{min / max} & \quad c^T x \quad \text{(primal program)} \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*} \]

\[ \begin{align*}
\text{max / min} & \quad b^T y \quad \text{(dual program)} \\
\text{subject to} & \quad A^T y \leq c \quad \text{no non-negativity restriction!}
\end{align*} \]
General Rules for Writing Dual Programs

<table>
<thead>
<tr>
<th>Maximization problem</th>
<th>Minimization problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints</td>
<td>Variables</td>
</tr>
<tr>
<td>$i$th constraint $\leq$</td>
<td>$i$th variable $\geq 0$</td>
</tr>
<tr>
<td>$i$th constraint $\geq$</td>
<td>$i$th variable $\leq 0$</td>
</tr>
<tr>
<td>$i$th constraint $=$</td>
<td>$i$th variable unrestricted</td>
</tr>
<tr>
<td>Variables</td>
<td>Constraints</td>
</tr>
<tr>
<td>$j$th variable $\geq 0$</td>
<td>$j$th constraint $\geq$</td>
</tr>
<tr>
<td>$j$th variable $\leq 0$</td>
<td>$j$th constraint $\leq$</td>
</tr>
<tr>
<td>$j$th variable unrestricted</td>
<td>$j$th constraint $=$</td>
</tr>
</tbody>
</table>

Table: Rules for converting between primals and duals.

Note
The dual of the dual is the primal!

Weak Duality – Canonical Form Version

Consider the following primal-dual pair in canonical form

**Primal LP:** \( \min \{ c^T x \mid Ax \geq b, x \geq 0 \} \),

**Dual LP:** \( \max \{ b^T y \mid A^T y \leq c, y \geq 0 \} \).

**Theorem (Weak Duality)**

*Suppose* \( x \) *is primal feasible, and* \( y \) *is dual feasible for the LPs defined above, then* \( c^T x \geq b^T y \).

**Corollary**

*If* \( x^* \) *is an primal-optimal and* \( y^* \) *is an dual-optimal, then* \( c^T x^* \geq b^T y^* \).

**Corollary**

*If* \( x^* \) *is primal-feasible,* \( y^* \) *is dual-feasible, and* \( c^T x^* = b^T y^* \), *then* \( x^* \) *and* \( y^* \) *are optimal for their respective programs.*
Consider the following primal-dual pair in standard form

**Primal LP:** \( \min \{ c^T x \mid Ax = b, x \geq 0 \} \),

**Dual LP:** \( \max \{ b^T y \mid A^T y \leq c \} \).

**Theorem (Weak Duality)**

*Suppose \( x \) is primal feasible, and \( y \) is dual feasible for the LPs defined above, then \( c^T x \geq b^T y \).*

**Corollary**

*If \( x^* \) is a primal-optimal and \( y^* \) is a dual-optimal, then \( c^T x^* \geq b^T y^* \).*

**Corollary**

*If \( x^* \) is primal-feasible, \( y^* \) is dual-feasible, and \( c^T x^* = b^T y^* \), then \( x^* \) and \( y^* \) are optimal for their respective programs.*

---

**Strong Duality**

**Theorem (Strong Duality)**

*If the primal LP has an optimal solution \( x^* \), then the dual LP has an optimal solution \( y^* \) such that*

\[ c^T x^* = b^T y^* . \]

---

**Primal**

<table>
<thead>
<tr>
<th>Feasible</th>
<th>Optimal</th>
<th>Unbounded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasible</td>
<td>X</td>
<td>Nah</td>
</tr>
<tr>
<td>Unbounded</td>
<td>Nah</td>
<td>X</td>
</tr>
<tr>
<td>Infeasible</td>
<td>Nah</td>
<td>X</td>
</tr>
</tbody>
</table>

**Dual**

<table>
<thead>
<tr>
<th>Feasible</th>
<th>Optimal</th>
<th>Infeasible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>X</td>
<td>Nah</td>
</tr>
<tr>
<td>Unbounded</td>
<td>Nah</td>
<td>X</td>
</tr>
</tbody>
</table>
The Diet Problem Revisited

The dual program for the diet problem:

\[
\begin{align*}
\text{max} & \quad b_1 y_1 + b_2 y_2 + \cdots + b_m y_m \\
\text{subject to} & \quad a_{11} y_1 + a_{21} y_2 + \cdots + a_{m1} y_m \geq c_1 \\
& \quad a_{12} y_1 + a_{22} y_2 + \cdots + a_{m2} y_m \geq c_2 \\
& \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
& \quad a_{1n} y_1 + a_{2n} y_2 + \cdots + a_{mn} y_m \geq c_n \\
& \quad y_j \geq 0, \forall j = 1, \ldots, m,
\end{align*}
\]

(Possible) Interpretation: \( y_i \) is the price per unit of nutrient \( i \) that a whole-seller sets to “manufacture” different types of foods.

The Max-Flow Problem Revisited

The dual program for the Max-Flow LP Formulation:

\[
\begin{align*}
\text{min} & \quad \sum_{uv \in E} c_{uv} y_{uv} \\
\text{subject to} & \quad y_{uv} - z_u + z_v \geq 0 \quad \forall uv \in E \\
& \quad z_s = 1 \\
& \quad z_t = 0 \\
& \quad y_{uv} \geq 0 \quad \forall uv \in E
\end{align*}
\]

Theorem (Max-Flow Min-Cut)

Maximum flow value equal minimum cut capacity.

Proof.

Let \((y^*, z^*)\) be optimal to the dual above. Set \(W = \{v \mid z_v^* \geq 1\}\), then total flow out of \(W\) is equal to \(\text{cap}(W, \overline{W})\).
Consider a linear program \( \min \{ c^T x \mid x \in P \} \), \( P \) is a polyhedron

1. Find a vertex of \( P \), if \( P \) is not empty (the LP is feasible)
2. Find a neighboring vertex with better cost
   - If found, then repeat step 2
   - Otherwise, either report UNBOUNDED or OPTIMAL SOLUTION

Questions

- When is \( P \) not empty?
- When does \( P \) have a vertex? (i.e. \( P \) is pointed)
- What is a vertex, anyhow?
- How to find an initial vertex?
- What if no vertex is optimal?
- How to find a “better” neighboring vertex
- Will the algorithm terminate?
- How long does it take?
3. What is a vertex, anyhow?

Many ways to define a vertex $v$:

- $v \in P$ a vertex iff $\exists y \neq 0$ with $v + y, v - y \in P$
- $v \in P$ a vertex iff $\exists u \neq w$ such that $v = (u + w)/2$
- $v \in P$ a vertex iff it’s the unique intersection of $n$ independent faces

Questions

- When is $P$ not empty?
- When does $P$ have a vertex? (i.e. $P$ is pointed)
- What is a vertex, anyhow?
- How to find an initial vertex?
- What if no vertex is optimal?
- How to find a “better” neighboring vertex
- Will the algorithm terminate?
- How long does it take?
2. When is $P$ pointed?

**Question**
Define a polyhedron which has no vertex?

**Lemma**
$P$ is pointed iff it contains no line

**Lemma**
$P = \{ x \mid Ax = b, x \geq 0 \}$, if not empty, always has a vertex.

**Lemma**
$v \in P = \{ x \mid Ax = b, x \geq 0 \}$ is a vertex iff the columns of $A$ corresponding to non-zero coordinates of $v$ are linearly independent

**Questions**
- When is $P$ not empty?
- When does $P$ have a vertex? (i.e. $P$ is pointed)
- What is a vertex, anyhow?
- How to find an initial vertex?
- What if no vertex is optimal?
- How to find a “better” neighboring vertex
- Will the algorithm terminate?
- How long does it take?
5. What if no vertex is optimal?

Lemma
Let $P = \{x \mid Ax = b, x \geq 0\}$. If $\min \{c^T x \mid x \in P\}$ is bounded (i.e. it has an optimal solution), then for all $x \in P$, there is a vertex $v \in P$ such that $c^T v \leq c^T x$.

Theorem
The linear program $\min \{c^T x \mid Ax = b, x \geq 0\}$ either

1. is infeasible,
2. is unbounded, or
3. has an optimal solution at a vertex.

Questions
1. When is $P$ not empty?
2. When does $P$ have a vertex? (i.e. $P$ is pointed)
3. What is a vertex, anyhow?
4. How to find an initial vertex?
5. What if no vertex is optimal?
6. How to find a “better” neighboring vertex
7. Will the algorithm terminate?
8. How long does it take?
6. How to find a “better” neighboring vertex

- The answer is the core of the Simplex method
- This is basically one iteration of the method

Consider a concrete example:

\[
\begin{align*}
\text{max} & \quad 3x_1 + 2x_2 + 4x_3 \\
\text{subject to} & \quad x_1 + x_2 + 2x_3 \leq 4 \\
& \quad 2x_1 + 3x_3 \leq 5 \\
& \quad 4x_1 + x_2 + 3x_3 \leq 7 \\
& \quad x_1, x_2, x_3 \geq 0.
\end{align*}
\]

Sample execution of the Simplex algorithm

Converting to standard form

\[
\begin{align*}
\text{max} & \quad 3x_1 + 2x_2 + 4x_3 \\
\text{subject to} & \quad x_1 + x_2 + 2x_3 + x_4 = 4 \\
& \quad 2x_1 + 3x_3 + x_5 = 5 \\
& \quad 4x_1 + x_2 + 3x_3 + x_6 = 7 \\
& \quad x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.
\end{align*}
\]

- \( \mathbf{x} = [0 \ 0 \ 4 \ 5 \ 7]^T \) is a vertex!
- Define \( B = \{4, 5, 6\}, \ N = \{1, 2, 3\} \).
- The variables \( x_i, i \in N \) are called free variables.
- The \( x_i \) with \( i \in B \) are basic variables.
- How does one improve \( \mathbf{x} \)? Increase \( x_3 \) as much as possible! (\( x_1 \) or \( x_2 \) works too.)
Sample execution of the Simplex algorithm

- $x_3$ can only be at most $5/3$, forcing $x_4 = 2/3, x_5 = 0, x_5 = 2$
- $x^T = [0 \ 0 \ 5/3 \ 2/3 \ 0 \ 2]$ is the new vertex (why?!!!)
- The new objective value is $20/3$
- $x_3$ enters the basis $B$, $x_5$ leaves the basis
- $B = \{3, 4, 6\}$, $N = \{1, 2, 5\}$

Rewrite the linear program

\[
\begin{align*}
\text{max} & \quad 3x_1 + 2x_2 + 4x_3 \\
\text{subject to} & \quad -\frac{1}{3}x_1 + x_2 + x_4 = \frac{2}{3} \\
& \quad \frac{2}{3}x_1 + x_3 + \frac{1}{3}x_5 = \frac{5}{3} \\
& \quad 2x_1 + x_2 + x_6 = 2
\end{align*}
\]

$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.$

We also want the objective function to depend only on the free variables:

\[
egin{align*}
3x_1 + 2x_2 + 4x_3 &= 3x_1 + 2x_2 + 4\left(\frac{5}{3} - \frac{2}{3}x_1 - \frac{1}{3}x_5\right) \\
&= \frac{1}{3}x_1 + 2x_2 - \frac{4}{3}x_5 + \frac{20}{3}
\end{align*}
\]

The linear program is thus equivalent to

\[
\begin{align*}
\text{max} & \quad \frac{1}{3}x_1 + 2x_2 + x_4 + \frac{20}{3} \\
\text{subject to} & \quad -\frac{1}{3}x_1 + x_2 + x_4 = \frac{2}{3} \\
& \quad \frac{2}{3}x_1 + x_3 + \frac{1}{3}x_5 = \frac{5}{3} \\
& \quad 2x_1 + x_2 + x_6 = 2
\end{align*}
\]

$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.$

Increase $x_2$ to $2/3$, so that $x_2$ enters, $x_4$ leaves.
Sample execution of the Simplex algorithm

\[
\begin{align*}
\text{max} & \quad x_1 - 2x_4 + 8 \\
\text{subject to} & \quad -\frac{1}{3}x_1 + x_2 + x_4 = \frac{2}{3} \\
& \quad \frac{2}{3}x_1 + x_3 + \frac{1}{3}x_5 = \frac{5}{3} \\
& \quad \frac{7}{3}x_1 - x_4 - \frac{1}{3}x_5 + x_6 = \frac{4}{3} \\
& \quad x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.
\end{align*}
\]

At this point, only \(x_1\) to increase.

- If all its coefficients are non-positive (like \(-1/3\) above), then the LP is UNBOUNDED.
- Fortunately, this is not the case here.
- Increase \(x_1\) to \(4/7\), so that \(x_1\) enters, \(x_6\) leaves.

Sample execution of the Simplex algorithm

\[
\begin{align*}
\text{max} & \quad -\frac{11}{7}x_4 + \frac{1}{7}x_5 - \frac{3}{7}x_6 + \frac{60}{7} \\
\text{subject to} & \quad +x_2 + \frac{6}{7}x_4 - \frac{5}{7}x_5 + \frac{1}{7}x_6 = \frac{6}{7} \\
& \quad +x_3 + \frac{2}{7}x_4 + \frac{3}{7}x_5 - \frac{2}{7}x_6 = \frac{9}{7} \\
& \quad -\frac{3}{7}x_4 - \frac{1}{7}x_5 + \frac{3}{7}x_6 = \frac{4}{7} \\
& \quad x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.
\end{align*}
\]

Now, \(x_5\) enters again, \(x_3\) leaves.
Sample execution of the Simplex algorithm

\[
\begin{align*}
\text{max} & \quad -\frac{1}{3}x_3 - \frac{34}{21}x_4 - \frac{1}{3}x_6 + 9 \\
\text{subject to} & \quad +x_2 + \frac{49}{15}x_3 + \frac{188}{105}x_4 - \frac{1}{3}x_6 = 3 \\
& \quad +\frac{7}{3}x_3 + \frac{2}{3}x_4 + x_5 - \frac{2}{3}x_6 = 3 \\
& \quad +\frac{1}{3}x_3 - \frac{1}{3}x_4 + \frac{1}{3}x_6 = 1 \\
& \quad x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.
\end{align*}
\]

Yeah! No more improvement is possible. We have reached the optimal vertex

\[v = \begin{bmatrix} 1 & 3 & 0 & 0 & 3 & 0 \end{bmatrix}^T.\]

The optimal cost is 9.

Questions

1. When is \( P \) not empty?
2. When does \( P \) have a vertex? (i.e. \( P \) is pointed)
3. What is a vertex, anyhow?
4. How to find an initial vertex?
5. What if no vertex is optimal?
6. How to find a “better” neighboring vertex
7. Will the algorithm terminate?
8. How long does it take?
Termination
- There are finitely many vertices \( \leq \binom{n}{m} \)
- Terminating = non-cycling, i.e. never come back to a vertex
- Many cycling prevention methods: perturbation method, lexicographic rule, Bland’s pivoting rule, etc.
  - Bland’s pivoting rule: pick smallest possible \( j \) to leave the basis, then smallest possible \( i \) to enter the basis

Running time
- Klee & Minty (1969) showed that Simplex could take exponential time

Summary: Simplex with Bland’s Rule
1. Start from a vertex \( v \) of \( P \).
2. Determine \( B \) and \( N \); Let \( y_B^T = c_B^T A_B^{-1} \).
3. If \( (c_N^T - y_B^T a_j) \geq 0 \), then vertex \( v \) is optimal. Moreover,
   \[
   c^T v = c_B^T v_B + c_N^T v_N = c_B^T (A_B^{-1} b - A_B^{-1} A_N v_N) + c_N^T v_N
   \]
   Else, let \( j = \min \{ j' \in N : (c_{j'} - y_B^T a_{j'}) < 0 \} \).
4. If \( A_B^{-1} a_j \leq 0 \), then report unbounded LP and STOP!
5. Otherwise, pick smallest \( k \in B \) such that \( (A_B^{-1} a_j)_k > 0 \) and that
   \[
   \frac{(A_B^{-1} b)_k}{(A_B^{-1} a_j)_k} = \min \left\{ \frac{(A_B^{-1} b)_i}{(A_B^{-1} a_j)_i} : i \in B, \ (A_B^{-1} a_j)_i > 0 \right\}.
   \]
6. \( x_k \) leaves, \( x_j \) enters: \( B = B \cup \{ j \} - \{ k \}, \ N = N \cup \{ k \} - \{ j \} \).
   Go back to step 3.
By Product: Strong Duality

Theorem (Strong Duality)

*If the primal LP has an optimal solution* \( x^* \), *then the dual LP has an optimal solution* \( y^* \) *such that*

\[
c^T x^* = b^T y^*.
\]

**Proof.**

- Suppose Simplex returns vertex \( x^* \) (at \( B \) and \( N \))
- Recall \( y_B^T = c_B^T A_B^{-1} \), then \( c^T x^* = y_B^T b \)

\[
A^T y_B = \begin{bmatrix} A_B^T \\ A_N^T \end{bmatrix} y_B = \begin{bmatrix} c_B \\ c_N \end{bmatrix} y_B 
\leq \begin{bmatrix} c_B \\ c_N \end{bmatrix} = c.
\]
- Set \( y^* = y_B^T \). Done!

Questions

- When is \( P \) not empty?
- When does \( P \) have a vertex? (i.e. \( P \) is pointed)
- What is a vertex, anyhow?
- How to find an initial vertex?
- What if no vertex is optimal?
- How to find a “better” neighboring vertex
- Will the algorithm terminate?
- How long does it take?
In $P = \{x \mid Ax = b, x \geq 0\}$, we can assume $b \geq 0$ (why?).

Let $A' = [A \quad I]$

Let $P' = \{z \mid A'z = b, z \geq 0\}$.

A vertex of $P'$ is $z = [0, \ldots, 0, b_1, \ldots, b_m]^T$.

$P$ is feasible iff the following LP has optimum value 0

$$
\min \left\{ \sum_{i=1}^{m} z_{n+i} \mid z \in P' \right\}
$$

From an optimal vertex $z^*$, ignore the last $m$ coordinates to obtain a vertex of $P$.

---

**Questions**

1. When is $P$ not empty?
2. When does $P$ have a vertex? (i.e. $P$ is pointed)
3. What is a vertex, anyhow?
4. How to find an initial vertex?
5. What if no vertex is optimal?
6. How to find a “better” neighboring vertex
7. Will the algorithm terminate?
8. How long does it take?