Linear Programming

Lecture 18
October 29, 2013
Part I

Linear Programming
Economic planning

Guns/nuclear-bombs/napkins/star-wars/professors/butter/mice problem

1. Penguina: a country.
2. Ruler need to decide how to allocate resources.
3. Maximize benefit.
4. Budget allocation
   (i) Nuclear bomb has a tremendous positive effect on security while being expensive.
   (ii) Guns, on the other hand, have a weaker effect.
5. Penguina need to prove a certain level of security:
   \[ x_{\text{gun}} + 1000 \times x_{\text{nuclear-bomb}} \geq 1000, \]
   where \( x_{\text{guns}} \): \# guns \( x_{\text{nuclear-bomb}} \): \# nuclear-bombs constructed.
6. \[ 100 \times x_{\text{gun}} + 1000000 \times x_{\text{nuclear-bomb}} \leq x_{\text{security}} \]
   \( x_{\text{security}} \): total amount spent on security.
   \( 100/1,000,000 \): price of producing a single gun/nuclear bomb.
An instance of **linear programming** (LP):

1. \( x_1, \ldots, x_n \): variables.

2. For \( j = 1, \ldots, m \): \( a_{j1} x_1 + \cdots + a_{jn} x_n \leq b_j \): linear inequality.

3. i.e., **constraint**.

4. Q: \( \exists s \) an assignment of values to \( x_1, \ldots, x_n \) such that all inequalities are satisfied.

5. Many possible solutions... Want solution that maximizes some linear quantity.

6. **objective function**: linear inequality being maximized.
Linear programming – example

\[
\begin{align*}
  a_{11}x_1 + \ldots + a_{1n}x_n & \leq b_1 \\
  a_{21}x_1 + \ldots + a_{2n}x_n & \leq b_2 \\
  \ldots \\
  a_{m1}x_1 + \ldots + a_{mn}x_n & \leq b_m \\
  \max c_1x_1 + \ldots + c_nx_n.
\end{align*}
\]
History

1. 1939: L. V. Kantorovich noticed the importance of certain type of Linear Programming problems for resource allocation.

2. 1947: Dantzig invented the simplex method for solving LP problems for the US Air force planning problems.

3. 1947: T. C. Koopmans showed LP provide the right model for the analysis of classical economic theories.

4. 1975: Koopmans and Kantorovich got the Nobel prize of economics.

5. Kantorovich the only the Russian economist that got the Nobel prize
Network flow via linear programming

Input: $G = (V, E)$ with source $s$ and sink $t$, and capacities $c(\cdot)$ on the edges. Compute max flow in $G$.

\[
\forall (u \rightarrow v) \in E \quad 0 \leq x_{u \rightarrow v} \quad x_{u \rightarrow v} \leq c(u \rightarrow v)
\]

\[
\forall v \in V \setminus \{s, t\} \quad \sum_{(u \rightarrow v) \in E} x_{u \rightarrow v} - \sum_{(v \rightarrow w) \in E} x_{v \rightarrow w} \leq 0
\]

\[
\sum_{(u \rightarrow v) \in E} x_{u \rightarrow v} - \sum_{(v \rightarrow w) \in E} x_{v \rightarrow w} \geq 0
\]

maximizing $\sum_{(s \rightarrow u) \in E} x_{s \rightarrow u}$
Part II

The Simplex Algorithm
Rewriting an LP

\[
\begin{align*}
\text{max} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \ldots, m
\end{align*}
\]

1. Rewrite: so every variable is non-negative.
2. Replace variable \( x_i \) by \( x'_i \) and \( x''_i \), where new constraints are:
   \[
x_i = x'_i - x''_i, \quad x'_i \geq 0 \quad \text{and} \quad x''_i \geq 0.
\]
3. Example: The (silly) LP \( 2x + y \geq 5 \) rewritten:
   \[
   2x' - 2x'' + y' - y'' \geq 5,
   \]
   \[
x' \geq 0, \quad y' \geq 0,
   \]
   \[
x'' \geq 0, \quad \text{and}
   \]
   \[
y'' \geq 0.
\]
Rewriting an LP

\[
\begin{align*}
\text{max} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \ldots, m
\end{align*}
\]

1. Rewrite: so every variable is non-negative.

2. Replace variable \( x_i \) by \( x_i' \) and \( x_i'' \), where new constraints are:
   \[ x_i = x_i' - x_i'', \quad x_i' \geq 0 \quad \text{and} \quad x_i'' \geq 0. \]

3. Example: The (silly) LP \( 2x + y \geq 5 \) rewritten:
   \[ 2x' - 2x'' + y' - y'' \geq 5, \]
   \[ x' \geq 0, \quad y' \geq 0, \]
   \[ x'' \geq 0, \quad \text{and} \]
   \[ y'' \geq 0. \]
Rewriting an LP

\[
\begin{align*}
\text{max} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \ldots, m
\end{align*}
\]

1. Rewrite: so every variable is non-negative.
2. Replace variable \( x_i \) by \( x_i' \) and \( x_i'' \), where new constraints are:
   \[ x_i = x_i' - x_i'', \quad x_i' \geq 0 \quad \text{and} \quad x_i'' \geq 0. \]
3. Example: The (silly) LP \( 2x + y \geq 5 \) rewritten:
   \[ 2x' - 2x'' + y' - y'' \geq 5, \]
   \[ x' \geq 0, \quad y' \geq 0, \]
   \[ x'' \geq 0, \quad \text{and} \quad y'' \geq 0. \]
Rewriting an LP into standard form

Lemma

Given an instance $I$ of LP, one can rewrite it into an equivalent LP, such that all the variables must be non-negative. This takes linear time in the size of $I$.

An LP where all variables must be non-negative is in standard form.
Rewriting an LP into standard form

Lemma

Given an instance $I$ of LP, one can rewrite it into an equivalent LP, such that all the variables must be non-negative. This takes linear time in the size of $I$.

An LP where all variables must be non-negative is in standard form.
A linear program in standard form.

\[
\begin{align*}
\text{max} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \ldots, m \\
& \quad x_j \geq 0 \quad \text{for } j = 1, \ldots, n.
\end{align*}
\]
Standard form of LP

Because everything is clearer when you use matrices. Not.

\[ A = \begin{pmatrix}
  a_{11} & a_{12} & \cdots & a_{1(n-1)} & a_{1n} \\
  a_{21} & a_{22} & \cdots & a_{2(n-1)} & a_{2n} \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  a_{(m-1)1} & a_{(m-1)2} & \cdots & a_{(m-1)(n-1)} & a_{(m-1)n} \\
  a_{m1} & a_{m2} & \cdots & a_{m(n-1)} & a_{mn}
\end{pmatrix}, \quad (\text{matrix notation.})
\]

\[ x \text{ is vector of unknowns.} \]
\[ \text{Solve LP for } x. \]

**LP in standard form.**

(Math notation.)

\[
\begin{align*}
\max \quad & c^T x \\
\text{s.t.} \quad & Ax \leq b. \\
\quad & x \geq 0.
\end{align*}
\]

\[ c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} \]
Slack Form

1. Next rewrite LP into slack form.
2. Every inequality becomes equality.
3. All variables must be positive.
4. See resulting form on the right.

New slack variables. Rewrite inequality: \( \sum_{i=1}^{n} a_i x_i \leq b \). As:

\[
\begin{align*}
  x_{n+1} &= b - \sum_{i=1}^{n} a_i x_i \\
  x_{n+1} &\geq 0.
\end{align*}
\]

Value of slack variable \( x_{n+1} \) encodes how far is the original inequality for holding with equality.
Slack Form

1. Next rewrite LP into \textit{slack form}.

2. Every inequality becomes equality.

3. All variables must be positive.

4. See resulting form on the right.

1. New \textit{slack variables}. Rewrite inequality: \( \sum_{i=1}^{n} a_i x_i \leq b \). As:

\[
\begin{align*}
x_{n+1} &= b - \sum_{i=1}^{n} a_i x_i \\
x_{n+1} &\geq 0.
\end{align*}
\]

2. Value of slack variable \( x_{n+1} \) encodes how far is the original inequality for holding with equality.
Slack Form

1. Next rewrite LP into **slack form**.
2. Every inequality becomes equality.
3. All variables must be positive.
4. See resulting form on the right.

New **slack variables**. Rewrite inequality: \( \sum_{i=1}^{n} a_i x_i \leq b \). As:

\[
\begin{align*}
    x_{n+1} &= b - \sum_{i=1}^{n} a_i x_i \\
    x_{n+1} &\geq 0.
\end{align*}
\]

Value of slack variable \(x_{n+1}\) encodes how far is the original inequality for holding with equality.
Slack Form

1. Next rewrite LP into **slack form**.
2. Every inequality becomes equality.
3. All variables must be positive.
4. See resulting form on the right.

New **slack variables**. Rewrite inequality: \( \sum_{i=1}^{n} a_i x_i \leq b \). As:

\[
x_{n+1} = b - \sum_{i=1}^{n} a_i x_i
\]

\[
x_{n+1} \geq 0.
\]

2. Value of slack variable \( x_{n+1} \) encodes how far is the original inequality for holding with equality.
1. LP now made of equalities of the form:
   \[ x_{n+1} = b - \sum_{i=1}^{n} a_i x_i \]

2. Variables on left: basic variables.

3. Variables on right: nonbasic variables.

4. LP in this form is in slack form.

Linear program in slack form.

\[ \text{max} \quad z = v + \sum_{j \in N} c_j x_j, \]
\[ \text{s.t.} \quad x_i = b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for} \quad i \in B, \]
\[ x_i \geq 0, \quad \forall i = 1, \ldots, n + m. \]
**Slack form...**

1. **LP** now made of equalities of the form:
   \[ x_{n+1} = b - \sum_{i=1}^{n} a_i x_i \]

2. Variables on left: **basic variables**.

3. Variables on right: **nonbasic variables**.

4. **LP** in this form is in **slack form**.

---

Linear program in slack form.

\[
\begin{align*}
\text{max} & \quad z = v + \sum_{j \in N} c_j x_j, \\
\text{s.t.} & \quad x_i = b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for} \quad i \in B, \\
& \quad x_i \geq 0, \quad \forall i = 1, \ldots, n + m.
\end{align*}
\]
Slack form...

1. **LP** now made of equalities of the form:
   \[ x_{n+1} = b - \sum_{i=1}^{n} a_i x_i \]

2. Variables on left: **basic variables**.

3. Variables on right: **nonbasic variables**.

4. **LP** in this form is in **slack form**.

---

Linear program in slack form.

\[
\begin{align*}
\text{max} & \quad z = v + \sum_{j \in N} c_j x_j, \\
\text{s.t.} & \quad x_i = b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for} \quad i \in B, \\
& \quad x_i \geq 0, \quad \forall i = 1, \ldots, n + m.
\end{align*}
\]
**Slack form...**

1. LP now made of equalities of the form:

   \[ x_{n+1} = b - \sum_{i=1}^{n} a_i x_i \]

2. Variables on left: **basic variables**.

3. Variables on right: **nonbasic variables**.

4. LP in this form is in **slack form**.

---

**Linear program in slack form.**

\[
\begin{align*}
\text{max} \quad & z = v + \sum_{j \in N} c_j x_j, \\
\text{s.t.} \quad & x_i = b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for} \quad i \in B, \\
& x_i \geq 0, \quad \forall i = 1, \ldots, n + m.
\end{align*}
\]
Slack form formally

Because everything is clearer when you use tuples. Not.

The slack form is defined by a tuple \((N, B, A, b, c, v)\).

- **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** = \(\{1, \ldots, n + m\}\)
- **v** - objective function constant.
Max \[ z = v + \sum_{j \in N} c_j x_j, \]

s.t. \[ x_i = b_i - \sum_{j \in N} a_{ij} x_j \text{ for } i \in B, \]

\[ x_i \geq 0, \quad \forall i = 1, \ldots, n + m. \]
Example

Consider the following LP which is in slack form.

\[
\begin{align*}
\text{max} \quad z &= 29 - \frac{1}{9}x_3 - \frac{1}{9}x_5 - \frac{2}{9}x_6 \\
x_1 &= 8 + \frac{1}{6}x_3 + \frac{1}{6}x_5 - \frac{1}{3}x_6 \\
x_2 &= 4 - \frac{1}{3}x_3 - \frac{1}{3}x_5 + \frac{1}{3}x_6 \\
x_4 &= 18 - \frac{1}{2}x_3 + \frac{1}{2}x_5
\end{align*}
\]
Example

...translated into tuple form \((N, B, A, b, c, v)\).

\[ B = \{1, 2, 4\} , N = \{3, 5, 6\} \]

\[
A = \begin{pmatrix}
  a_{13} & a_{15} & a_{16} \\
  a_{23} & a_{25} & a_{26} \\
  a_{43} & a_{45} & a_{46}
\end{pmatrix} = \begin{pmatrix}
  -1/6 & -1/6 & 1/3 \\
  8/3 & 2/3 & -1/3 \\
  1/2 & -1/2 & 0
\end{pmatrix}
\]

\[
b = \begin{pmatrix}
  b_1 \\
  b_2 \\
  b_4
\end{pmatrix} = \begin{pmatrix}
  8 \\
  4 \\
  18
\end{pmatrix}
\]

\[
c = \begin{pmatrix}
  c_3 \\
  c_5 \\
  c_6
\end{pmatrix} = \begin{pmatrix}
  -1/9 \\
  -1/9 \\
  -2/9
\end{pmatrix}
\]

\[ v = 29. \]

Note that indices depend on the sets \(N\) and \(B\), and also that the entries in \(A\) are negation of what they appear in the slack form.
Another example...

\[
\begin{align*}
\text{max} & \quad 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} & \quad 2x_1 + 3x_2 + x_3 \leq 5 \\
& \quad 4x_1 + x_2 + 2x_3 \leq 11 \\
& \quad 3x_1 + 4x_2 + 2x_3 \leq 8 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]

Transform into slack form...

\[
\begin{align*}
\text{max} & \quad z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} & \quad w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& \quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& \quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& \quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]
The Simplex algorithm by example

**max** \[ 5x_1 + 4x_2 + 3x_3 \]

**s.t.** \[
\begin{align*}
2x_1 + 3x_2 + x_3 & \leq 5 \\
4x_1 + x_2 + 2x_3 & \leq 11 \\
3x_1 + 4x_2 + 2x_3 & \leq 8 \\
x_1, x_2, x_3 & \geq 0
\end{align*}
\]

Next, we introduce slack variables, for example, rewriting \( 2x_1 + 3x_2 + x_3 \leq 5 \) as the constraints: \( w_1 \geq 0 \) and \( w_1 = 5 - 2x_1 - 3x_2 - x_3 \). The resulting LP in slack form is

**max** \[ z = 5x_1 + 4x_2 + 3x_3 \]

**s.t.** \[
\begin{align*}
w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
w_2 &= 11 - 4x_1 - x_2 - 2x_3 \\
w_3 &= 8 - 3x_1 - 4x_2 - 2x_3 \\
x_1, x_2, x_3, w_1, w_2, w_3 & \geq 0
\end{align*}
\]
Example continued...

\[
\begin{align*}
\text{max } z &= 5x_1 + 4x_2 + 3x_3 \\
\text{s.t. } w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
&
\quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
&
\quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
&
\quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. \( w_1, w_2, w_3 \): slack variables.
   (Also currently basic variables).

2. Consider the slack representation trivial solution...
   all non-basic variables assigned zero:
   \( x_1 = x_2 = x_3 = 0 \).

3. \( \implies w_1 = 5, w_2 = 11 \) and \( w_3 = 8 \).

4. Feasible!

5. Objection function value: \( z = 0 \).

6. Further improve \( t \) value of objective function (i.e., \( z \)). While keeping feasibility.
Example continued...

\[
\begin{align*}
\text{max} & \quad z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} & \quad w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& \quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& \quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& \quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. \( w_1, w_2, w_3 \): slack variables. (Also currently basic variables).

2. Consider the slack representation trivial solution...
   all non-basic variables assigned zero:
   \( x_1 = x_2 = x_3 = 0 \).

3. \( \implies w_1 = 5, w_2 = 11 \) and \( w_3 = 8 \).

4. Feasible!

5. Objection function value: \( z = 0 \).

4. Further improve the value of objective function (i.e., \( z \)). While keeping feasibility.
Example continued...

\[
\begin{align*}
\text{max} \quad z &= 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
&= 11 - 4x_1 - x_2 - 2x_3 \\
&= 8 - 3x_1 - 4x_2 - 2x_3 \\
x_1, x_2, x_3, w_1, w_2, w_3 &\geq 0
\end{align*}
\]

1. \( w_1, w_2, w_3 \): slack variables. (Also currently basic variables).

2. Consider the slack representation trivial solution...
   all non-basic variables assigned zero:
   \( x_1 = x_2 = x_3 = 0 \).

3. \( \Rightarrow \quad w_1 = 5, w_2 = 11 \) and \( w_3 = 8 \).

4. Feasible!

5. Objection function value: \( z = 0 \).

6. Further improve t value of objective function (i.e., \( z \)). While keeping feasibility.
Example continued...

\[
\begin{align*}
\text{max} \quad z &= 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
&\quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
&\quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
&\quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. \(w_1, w_2, w_3\): slack variables. (Also currently basic variables).

2. Consider the slack representation trivial solution... all non-basic variables assigned zero:
   \(x_1 = x_2 = x_3 = 0\).

3. \(\iff\) \(w_1 = 5, w_2 = 11\) and \(w_3 = 8\).

4. Feasible!

5. Objection function value: \(z = 0\).

6. Further improve t value of objective function (i.e., \(z\)). While keeping feasibility.
Example continued...

\[ \text{max } z = 5x_1 + 4x_2 + 3x_3 \]
\[ \text{s.t. } w_1 = 5 - 2x_1 - 3x_2 - x_3 \]
\[ w_2 = 11 - 4x_1 - x_2 - 2x_3 \]
\[ w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \]
\[ x_1, x_2, x_3, w_1, w_2, w_3 \geq 0 \]

1. \( w_1, w_2, w_3 \): slack variables. (Also currently basic variables).

2. Consider the slack representation trivial solution...
   all non-basic variables assigned zero:
   \[ x_1 = x_2 = x_3 = 0. \]

3. \( \Rightarrow w_1 = 5, w_2 = 11 \text{ and } w_3 = 8. \)

4. Feasible!

5. Objection function value: \( z = 0. \)

6. Further improve the value of objective function (i.e., \( z \)). While keeping feasibility.
Example continued...

\[
\begin{align*}
\text{max} & \quad z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} & \quad w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& \quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& \quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& \quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

\begin{enumerate}
\item \(w_1, w_2, w_3\): slack variables. (Also currently basic variables).
\item Consider the slack representation trivial solution...
all non-basic variables assigned zero:
\(x_1 = x_2 = x_3 = 0\).

\begin{array}{l}
\Rightarrow w_1 = 5, \, w_2 = 11 \text{ and } w_3 = 8.
\end{array}

\item Feasible!

\item Objection function value: \(z = 0\).

\item Further improve \(t\) value of objective function (i.e., \(z\)). While keeping feasibility.
Example continued...

\[ \text{max} \quad z = 5x_1 + 4x_2 + 3x_3 \]
\[ \text{s.t.} \quad w_1 = 5 - 2x_1 - 3x_2 - x_3 \]
\[ w_2 = 11 - 4x_1 - x_2 - 2x_3 \]
\[ w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \]
\[ x_1, x_2, x_3, w_1, w_2, w_3 \geq 0 \]

1. \( w_1, w_2, w_3 \): slack variables.
   (Also currently basic variables).

2. Consider the slack representation trivial solution...
   all non-basic variables assigned zero:
   \( x_1 = x_2 = x_3 = 0 \).

3. \( \implies \) \( w_1 = 5, \ w_2 = 11 \) and \( w_3 = 8 \).

4. Feasible!

5. Objection function value: \( z = 0 \).

6. Further improve t value of objective function (i.e., \( z \)). While keeping feasibility.
Example continued...

\[
\begin{align*}
\text{max} \quad z &= 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
&\quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
&\quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
&\quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. \(x_1 = x_2 = x_3 = 0\) \(\implies w_1 = 5, w_2 = 11\) and \(w_3 = 8\).

2. All \(w_i\) positive – change \(x_i\) a bit does not change feasibility.

1. \(z = 5x_1 + 4x_2 + 3x_3\): want to increase values of \(x_1\)s... since \(z\) increases (since \(5 > 0\)).

2. How much to increase \(x_1\)???


4. Increase \(x_1\) as much as possible without breaking feasibility!
Example continued...

max \quad z = 5x_1 + 4x_2 + 3x_3
s.t. \quad w_1 = 5 - 2x_1 - 3x_2 - x_3
\quad w_2 = 11 - 4x_1 - x_2 - 2x_3
\quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3
\quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0

1. \quad x_1 = x_2 = x_3 = 0
   \quad \implies w_1 = 5, \ w_2 = 11
   \quad \text{and} \ w_3 = 8.

2. \quad \text{All } w_i \text{ positive – change } x_i \text{ a bit does not change feasibility.}

1. \quad z = 5x_1 + 4x_2 + 3x_3: \text{ want to increase values of } x_1 \text{s... since } z
   \text{ increases (since } 5 > 0).\
2. \quad \text{How much to increase } x_1 \text{??}
3. \quad \text{Careful! Might break feasibility.}
4. \quad \text{Increase } x_1 \text{ as much as possible without breaking feasibility!}
Example continued...

\[ \text{max } z = 5x_1 + 4x_2 + 3x_3 \]
\[ \text{s.t. } w_1 = 5 - 2x_1 - 3x_2 - x_3 \]
\[ w_2 = 11 - 4x_1 - x_2 - 2x_3 \]
\[ w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \]
\[ x_1, x_2, x_3, w_1, w_2, w_3 \geq 0 \]

1. \( x_1 = x_2 = x_3 = 0 \)
   \[ \implies w_1 = 5, w_2 = 11 \]
   and \( w_3 = 8. \)

2. All \( w_i \) positive – change \( x_i \) a bit does not change feasibility.

1. \( z = 5x_1 + 4x_2 + 3x_3 \): want to increase values of \( x_1 \)s... since \( z \) increases (since \( 5 > 0 \)).

2. How much to increase \( x_1 \)???


4. Increase \( x_1 \) as much as possible without breaking feasibility!
Example continued...

\[ \begin{align*}
\text{max} \quad z &= 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
\quad w_2 &= 11 - 4x_1 - x_2 - 2x_3 \\
\quad w_3 &= 8 - 3x_1 - 4x_2 - 2x_3 \\
\quad x_1, x_2, x_3, w_1, w_2, w_3 &\geq 0
\end{align*} \]

1. \[ x_1 = x_2 = x_3 = 0 \quad \implies \quad w_1 = 5, \ w_2 = 11 \quad \text{and} \quad w_3 = 8. \]

2. All \( w_i \) positive – change \( x_i \) a bit does not change feasibility.

1. \( z = 5x_1 + 4x_2 + 3x_3 \): want to increase values of \( x_1 \)s... since \( z \) increases (since 5 > 0).

2. How much to increase \( x_1 \)???


4. Increase \( x_1 \) as much as possible without breaking feasibility!
Example continued...

\[
\begin{align*}
\text{max} \quad z &= 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
&= 11 - 4x_1 - x_2 - 2x_3 \\
&= 8 - 3x_1 - 4x_2 - 2x_3 \\
x_1, x_2, x_3, w_1, w_2, w_3 &\geq 0
\end{align*}
\]

1. \(x_1 = x_2 = x_3 = 0\) \(\Rightarrow w_1 = 5, w_2 = 11\) and \(w_3 = 8\).

2. All \(w_i\) positive – change \(x_i\) a bit does not change feasibility.

1. \(z = 5x_1 + 4x_2 + 3x_3\): want to increase values of \(x_1\)s... since \(z\) increases (since 5 > 0).

2. How much to increase \(x_1\)???


4. Increase \(x_1\) as much as possible without breaking feasibility!
Example continued...

\[
\begin{align*}
\max \quad z &= 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
&\quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
&\quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
&\quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. \(x_1 = x_2 = x_3 = 0\) \(\implies w_1 = 5, w_2 = 11\) and \(w_3 = 8\).
2. All \(w_i\) positive – change \(x_i\) a bit does not change feasibility.

1. \(z = 5x_1 + 4x_2 + 3x_3\): want to increase values of \(x_1\)s... since \(z\) increases (since \(5 > 0\)).
2. How much to increase \(x_1\)???
4. Increase \(x_1\) as much as possible without breaking feasibility!
Example continued...

\[
\begin{align*}
\text{max} & \quad z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} & \quad w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& \quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& \quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& \quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. \[x_1 = x_2 = x_3 = 0 \quad \implies \quad w_1 = 5, \ w_2 = 11\] and \[w_3 = 8.\]

2. All \(w_i\) positive – change \(x_i\) a bit does not change feasibility.

\begin{enumerate}
\item \(z = 5x_1 + 4x_2 + 3x_3\): want to increase values of \(x_1\)s... since \(z\) increases (since \(5 > 0\)).
\item How much to increase \(x_1\) ???
\item Careful! Might break feasibility.
\item Increase \(x_1\) as much as possible without breaking feasibility!
\end{enumerate}
Example continued...

\[
\begin{align*}
\text{max } & \quad z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t. } & \quad w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& \quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& \quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& \quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. Set \( x_2 = x_3 = 0 \)

\[
\begin{align*}
& \quad w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& \quad \quad = 5 - 2x_1 \\
& \quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& \quad \quad = 11 - 4x_1 \\
& \quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& \quad \quad = 8 - 3x_1.
\end{align*}
\]

4. Want to increase \( x_1 \) as much as possible, as long as:

\[
\begin{align*}
& \quad w_1 = 5 - 2x_1 \geq 0, \\
& \quad w_2 = 11 - 4x_1 \geq 0, \\
& \text{and } w_3 = 8 - 3x_1 \geq 0.
\end{align*}
\]
Example continued...

\[ \text{max } z = 5x_1 + 4x_2 + 3x_3 \]
\[ \text{s.t. } w_1 = 5 - 2x_1 - 3x_2 - x_3 \]
\[ w_2 = 11 - 4x_1 - x_2 - 2x_3 \]
\[ w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \]
\[ x_1, x_2, x_3, w_1, w_2, w_3 \geq 0 \]

\[ \text{Set } x_2 = x_3 = 0 \]
\[ w_1 = 5 - 2x_1 \]
\[ = 5 - 2x_1 \]
\[ w_2 = 11 - 4x_1 \]
\[ = 11 - 4x_1 \]
\[ w_3 = 8 - 3x_1 \]
\[ = 8 - 3x_1. \]

Want to increase \( x_1 \) as much as possible, as long as:
\[ w_1 = 5 - 2x_1 \geq 0, \]
\[ w_2 = 11 - 4x_1 \geq 0, \]
and \( w_3 = 8 - 3x_1 \geq 0. \)
Example continued...

\[
\text{max } z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t. } w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\]

1 Set \( x_2 = x_3 = 0 \)

\[
w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
= 5 - 2x_1 \\
w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
= 11 - 4x_1 \\
w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
= 8 - 3x_1.
\]

4 Want to increase \( x_1 \) as much as possible, as long as:

\[
w_1 = 5 - 2x_1 \geq 0, \\
w_2 = 11 - 4x_1 \geq 0, \\
\text{and } w_3 = 8 - 3x_1 \geq 0.
\]
Example continued...

\[
\begin{aligned}
\text{max } \quad & z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t. } \quad & w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& x_1, x_2, x_3, w_1, w_2, w_3 \geq 0 \\
\end{aligned}
\]

1. **Constraints:**
   \[
   \begin{aligned}
   w_1 &= 5 - 2x_1 \geq 0, \\
   w_2 &= 11 - 4x_1 \geq 0, \\
   \text{and } w_3 &= 8 - 3x_1 \geq 0.
   \end{aligned}
   \]

2. Maximum we can increase \(x_1\) is 2.5.

3. \(x_1 = 2.5, \ x_2 = 0, \ x_3 = 0, \ w_1 = 0, \ w_2 = 1, \ w_3 = 0.5\)  
   \[\Rightarrow \quad z = 5x_1 + 4x_2 + 3x_3 = 12.5.\]

4. Improved target!

5. A nonbasic variable \(x_1\) is now non-zero. One basic variable \((w_1)\) became zero.
Example continued...

\[
\begin{align*}
\text{max } & \quad z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t. } & \quad w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& \quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& \quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& \quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. **Constraints:**

\[
\begin{align*}
w_1 &= 5 - 2x_1 \geq 0, \\
w_2 &= 11 - 4x_1 \geq 0, \\
\text{and } w_3 &= 8 - 3x_1 \geq 0.
\end{align*}
\]

2. **Maximum we can increase \(x_1\) is 2.5.**

3. \(x_1 = 2.5, \ x_2 = 0, \ x_3 = 0, \ w_1 = 0, \ w_2 = 1, \ w_3 = 0.5\)

   \[\Rightarrow \quad z = 5x_1 + 4x_2 + 3x_3 = 12.5.\]

4. **Improved target!**

5. **A nonbasic variable \(x_1\) is now non-zero. One basic variable \((w_1)\) became zero.**
Example continued...

\[
\text{max } z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t. } w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\]

1. Maximum we can increase \( x_1 \) is 2.5.

2. \( x_1 = 2.5, \ x_2 = 0, \ x_3 = 0, \ w_1 = 0, \ w_2 = 1, \ w_3 = 0.5 \)
   \[ \Rightarrow z = 5x_1 + 4x_2 + 3x_3 = 12.5. \]

3. Improved target!

4. A nonbasic variable \( x_1 \) is now non-zero. One basic variable \( (w_1) \) became zero.

Constraints:

\[
w_1 = 5 - 2x_1 \geq 0, \\
w_2 = 11 - 4x_1 \geq 0, \\
\text{and } w_3 = 8 - 3x_1 \geq 0.
\]

\[
x_1 \leq 2.5, \\
x_1 \leq 11/4 = 2.75 \text{ and } \\
x_1 \leq 8/3 = 2.66
\]
Example continued...

\[
\begin{align*}
\text{max} \quad z &= 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
&\quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
&\quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
&\quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. Maximum we can increase \(x_1\) is 2.5.
2. \(x_1 = 2.5, \ x_2 = 0, \ x_3 = 0, \ w_1 = 0, \ w_2 = 1, \ w_3 = 0.5\)
   \[\Rightarrow \quad z = 5x_1 + 4x_2 + 3x_3 = 12.5.\]
3. Improved target!
4. A nonbasic variable \(x_1\) is now non-zero. One basic variable (\(w_1\)) became zero.

1. Constraints:
   \[
   \begin{align*}
   w_1 &= 5 - 2x_1 \geq 0, \\
   w_2 &= 11 - 4x_1 \geq 0, \\
   \text{and} \quad w_3 &= 8 - 3x_1 \geq 0.
   \end{align*}
   \]

2. \(x_1 \leq 2.5, \quad x_1 \leq 11/4 = 2.75 \) and \(x_1 \leq 8/3 = 2.66\)
Example continued...

\[
\begin{align*}
\text{max} \quad z &= 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
& \quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& \quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& \quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. Constraints:
   \[
   \begin{align*}
   w_1 &= 5 - 2x_1 \geq 0, \\
   w_2 &= 11 - 4x_1 \geq 0, \\
   \text{and } w_3 &= 8 - 3x_1 \geq 0.
   \end{align*}
   \]

2. Maximum we can increase \(x_1\) is 2.5.
   \[
   x_1 = 2.5, \quad x_2 = 0, \quad x_3 = 0, \quad w_1 = 0, \quad w_2 = 1, \quad w_3 = 0.5
   \Rightarrow \quad z = 5x_1 + 4x_2 + 3x_3 = 12.5.
   \]

3. Improved target!

4. A nonbasic variable \(x_1\) is now non-zero. One basic variable \((w_1)\) became zero.
Example continued...

\[
\begin{align*}
\text{max } z &= 5x_1 + 4x_2 + 3x_3 \\
\text{s.t. } w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\
w_2 &= 11 - 4x_1 - x_2 - 2x_3 \\
w_3 &= 8 - 3x_1 - 4x_2 - 2x_3 \\
x_1, x_2, x_3, w_1, w_2, w_3 &\geq 0
\end{align*}
\]

1. Maximum we can increase \( x_1 \) is 2.5.
2. \( x_1 = 2.5, \ x_2 = 0, \ x_3 = 0, \ w_1 = 0, \ w_2 = 1, \ w_3 = 0.5 \)  
   \[ \Rightarrow \quad z = 5x_1 + 4x_2 + 3x_3 = 12.5. \]
3. Improved target!
4. A nonbasic variable \( x_1 \) is now non-zero. One basic variable \( (w_1) \) became zero.

Constraints:

\[
\begin{align*}
w_1 &= 5 - 2x_1 \geq 0, \\
w_2 &= 11 - 4x_1 \geq 0, \\
\text{and } w_3 &= 8 - 3x_1 \geq 0.
\end{align*}
\]

\[
\begin{align*}
x_1 &\leq 2.5, \\
x_1 &\leq 11/4 = 2.75 \text{ and } \\
x_1 &\leq 8/3 = 2.66
\end{align*}
\]
Example continued...

\[
\begin{align*}
\text{max} & \quad z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} & \quad w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& \quad w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& \quad w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& \quad x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. \(x_1 = 2.5, \ x_2 = 0, \ x_3 = 0, \ w_1 = 0, \ w_2 = 1, \ w_3 = 0.5\)

2. A nonbasic variable \(x_1\) is now non-zero. One basic variable \((w_1)\) became zero.

3. Want to keep invariant: All non-basic variables in current solution are zero...

4. Idea: Exchange \(x_1\) and \(w_1\)!

5. Consider equality LP with \(w_1\) and \(x_1\).

\[
w_1 = 5 - 2x_1 - 3x_2 - x_3.
\]

6. Rewrite as: \(x_1 = 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3\).
Example continued...

$$\text{max} \quad z = 5x_1 + 4x_2 + 3x_3$$

$$\text{s.t.} \quad w_1 = 5 - 2x_1 - 3x_2 - x_3$$

$$w_2 = 11 - 4x_1 - x_2 - 2x_3$$

$$w_3 = 8 - 3x_1 - 4x_2 - 2x_3$$

$$x_1, x_2, x_3, w_1, w_2, w_3 \geq 0$$

1. \(x_1 = 2.5, \; x_2 = 0, \; x_3 = 0, \; w_1 = 0, \; w_2 = 1, \; w_3 = 0.5\)

2. A nonbasic variable \(x_1\) is now non-zero. One basic variable \(w_1\) became zero.

1. Want to keep invariant: All non-basic variables in current solution are zero...

2. Idea: Exchange \(x_1\) and \(w_1\)!

3. Consider equality LP with \(w_1\) and \(x_1\).

   \(w_1 = 5 - 2x_1 - 3x_2 - x_3.\)

4. Rewrite as: \(x_1 = 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3.\)
Example continued...

\[
\begin{align*}
\text{max} \quad & z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad & w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. \( x_1 = 2.5, \quad x_2 = 0, \quad x_3 = 0, \quad w_1 = 0, \quad w_2 = 1, \quad w_3 = 0.5 \)

2. A nonbasic variable \( x_1 \) is now non-zero. One basic variable (\( w_1 \)) became zero.

1. Want to keep invariant: All non-basic variables in current solution are zero...

2. Idea: Exchange \( x_1 \) and \( w_1 \)!

3. Consider equality LP with \( w_1 \) and \( x_1 \).
   \[ w_1 = 5 - 2x_1 - 3x_2 - x_3. \]

4. Rewrite as: \( x_1 = 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3. \)
Example continued...

\[
\begin{align*}
\max \quad & z = 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad & w_1 = 5 - 2x_1 - 3x_2 - x_3 \\
& w_2 = 11 - 4x_1 - x_2 - 2x_3 \\
& w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\
& x_1, x_2, x_3, w_1, w_2, w_3 \geq 0
\end{align*}
\]

1. \(x_1 = 2.5, \quad x_2 = 0, \quad x_3 = 0, \quad w_1 = 0, \quad w_2 = 1, \quad w_3 = 0.5\)

2. A nonbasic variable \(x_1\) is now non-zero. One basic variable \((w_1)\) became zero.

1. Want to keep invariant: All non-basic variables in current solution are zero...

2. Idea: Exchange \(x_1\) and \(w_1\)!

3. Consider equality LP with \(w_1\) and \(x_1\).
   \[w_1 = 5 - 2x_1 - 3x_2 - x_3.\]

4. Rewrite as: \(x_1 = 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3.\)
Example continued...

Substituting $x_1 = 5 - 2x_1 - 3x_2 - x_3$, the new LP

\[
\begin{align*}
\text{max} & \quad z = 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
x_1 &= 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
w_2 &= 1 + 2w_1 + 5x_2 \\
w_3 &= 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}
\]

1. nonbasic variables: $\{w_1, x_2, x_3\}$
2. basic variables: $\{x_1, w_2, w_3\}$.
3. Trivial solution: all nonbasic variables $= 0$ is feasible.
4. $w_1 = x_2 = x_3 = 0$. Value: $z = 12.5$. 

Sariel (UIUC) CS573 Fall 2013 26 / 35
Example continued...

Substituting $x_1 = 5 - 2x_1 - 3x_2 - x_3$, the new LP

\[
\begin{align*}
\text{max } z &= 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
x_1 &= 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
w_2 &= 1 + 2w_1 + 5x_2 \\
w_3 &= 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}
\]

1. nonbasic variables: $\{w_1, x_2, x_3\}$
   basic variables: $\{x_1, w_2, w_3\}$.

2. Trivial solution: all nonbasic variables $= 0$ is feasible.

3. $w_1 = x_2 = x_3 = 0$. Value: $z = 12.5$. 
Example continued...

Substituting \( x_1 = 5 - 2x_1 - 3x_2 - x_3 \), the new LP

\[
\begin{align*}
\text{max} & \quad z = 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
x_1 &= 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
w_2 &= 1 + 2w_1 + 5x_2 \\
w_3 &= 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}
\]

1 nonbasic variables: \( \{w_1, x_2, x_3\} \)
basic variables: \( \{x_1, w_2, w_3\} \).

2 Trivial solution: all nonbasic variables = 0 is feasible.

3 \( w_1 = x_2 = x_3 = 0 \). Value: \( z = 12.5 \).
Rewriting stop done is called **pivoting**.

**pivoted on** $x_1$.

Continue pivoting till reach optimal solution.

$$\begin{align*}
\text{max} \quad & z = 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
& x_1 = 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
& w_2 = 1 + 2w_1 + 5x_2 \\
& w_3 = 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}$$

Can not pivot on $w_1$, since if $w_1$ increase, then $z$ decreases. Bad.

Can not pivot on $x_2$ (coefficient in objective function is $-3.5$).

Can only pivot on $x_3$ since its coefficient ub objective 0.5. Positive number.
Example continued...

1. Rewriting stop done is called *pivot*ing.

2. pivoted on $x_1$.


   \[
   \begin{align*}
   \text{max } z &= 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
   x_1 &= 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
   w_2 &= 1 + 2w_1 + 5x_2 \\
   w_3 &= 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
   \end{align*}
   \]

4. Can not pivot on $w_1$, since if $w_1$ increase, then $z$ decreases. Bad.

5. Can not pivot on $x_2$ (coefficient in objective function is $-3.5$).

6. Can only pivot on $x_3$ since its coefficient ub objective 0.5. Positive number.
Example continued...

1. Rewriting stop done is called **pivoting**.
2. pivoted on $x_1$.

\[
\begin{align*}
\text{max} & \quad z = 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
x_1 &= 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
w_2 &= 1 + 2w_1 + 5x_2 \\
w_3 &= 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}
\]

4. Can not pivot on $w_1$, since if $w_1$ increase, then $z$ decreases. Bad.
5. Can not pivot on $x_2$ (coefficient in objective function is $-3.5$).
6. Can only pivot on $x_3$ since its coefficient ub objective 0.5. Positive number.
Example continued...

1. Rewriting stop done is called \textit{pivoting}.
2. Pivoted on $x_1$.

\[
\begin{align*}
\text{max} & \quad z = 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
x_1 &= 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
w_2 &= 1 + 2w_1 + 5x_2 \\
w_3 &= 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}
\]

4. Can not pivot on $w_1$, since if $w_1$ increase, then $z$ decreases. Bad.
5. Can not pivot on $x_2$ (coefficient in objective function is $-3.5$).
6. Can only pivot on $x_3$ since its coefficient ub objective 0.5. Positive number.
Example continued...

1. Rewriting stop done is called **pivoting**.

2. pivoted on $x_1$.


\[
\begin{align*}
\text{max} \quad z &= 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
x_1 &= 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
w_2 &= 1 + 2w_1 + 5x_2 \\
w_3 &= 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}
\]

4. Can not pivot on $w_1$, since if $w_1$ increase, then $z$ decreases. Bad.

5. Can not pivot on $x_2$ (coefficient in objective function is $-3.5$).

6. Can only pivot on $x_3$ since its coefficient ub objective 0.5. Positive number.
Example continued...

1. Rewriting stop done is called \textit{pivoting}.
2. pivoted on \( x_1 \).

\[
\begin{align*}
\text{max } z &= 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
x_1 &= 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
w_2 &= 1 + 2w_1 + 5x_2 \\
w_3 &= 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}
\]

4. Can not pivot on \( w_1 \), since if \( w_1 \) increase, then \( z \) decreases. Bad.

5. Can not pivot on \( x_2 \) (coefficient in objective function is \(-3.5\)).

6. Can only pivot on \( x_3 \) since its coefficient ub objective 0.5. Positive number.
Example continued...

1. Rewriting stop done is called *pivoting*.
2. pivoted on $x_1$.

\[
\begin{align*}
\max & \quad z = 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
x_1 & = 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
w_2 & = 1 + 2w_1 + 5x_2 \\
w_3 & = 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}
\]

4. Can not pivot on $w_1$, since if $w_1$ increase, then $z$ decreases. Bad.
5. Can not pivot on $x_2$ (coefficient in objective function is $-3.5$).
6. Can only pivot on $x_3$ since its coefficient ub objective 0.5. Positive number.
Example continued...

\[
\begin{align*}
\text{max} & \quad z = 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
x_1 & = 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
w_2 & = 1 + 2w_1 + 5x_2 \\
w_3 & = 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}
\]

1. Can only pivot on \( x_3 \)... 
2. \( x_1 \) can only be increased to 1 before \( w_3 = 0 \).
3. Rewriting the equality for \( w_3 \) in LP: 
   \[
   w_3 = 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3,
   \]
4. ...for \( x_3 \): \( x_3 = 1 + 3w_1 + x_2 - 2w_3 \).
5. Substituting into LP, we get the following LP.

\[
\begin{align*}
\text{max} & \quad z = 13 - w_1 - 3x_2 - w_3 \\
s.t. & \quad x_1 = 2 - 2w_1 - 2x_2 + w_3 \\
w_2 & = 1 + 2w_1 + 5x_2 \\
x_3 & = 1 + 3w_1 + x_2 - 2w_3
\end{align*}
\]
Example continued...

\[
\begin{align*}
\text{max} & \quad z = 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3 \\
x_1 & = 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3 \\
w_2 & = 1 + 2w_1 + 5x_2 \\
w_3 & = 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.
\end{align*}
\]

1. Can only pivot on \(x_3\)...
2. \(x_1\) can only be increased to 1 before \(w_3 = 0\).
3. Rewriting the equality for \(w_3\) in LP:
   \[w_3 = 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3,\]
4. ...for \(x_3\): \(x_3 = 1 + 3w_1 + x_2 - 2w_3\).
5. Substituting into LP, we get the following LP.

\[
\begin{align*}
\text{max} & \quad z = 13 - w_1 - 3x_2 - w_3 \\
\text{s.t.} & \quad x_1 = 2 - 2w_1 - 2x_2 + w_3 \\
&w_2 = 1 + 2w_1 + 5x_2 \\
x_3 & = 1 + 3w_1 + x_2 - 2w_3
\end{align*}
\]
Example continued – can this be further improved?

\[
\begin{align*}
\text{max} \quad z &= 13 - w_1 - 3x_2 - w_3 \\
\text{s.t.} \quad x_1 &= 2 - 2w_1 - 2x_2 + w_3 \\
&\quad w_2 = 1 + 2w_1 + 5x_2 \\
&\quad x_3 = 1 + 3w_1 + x_2 - 2w_3
\end{align*}
\]

1. NO!
2. All coefficients in objective negative (or zero).
3. Trivial solution (all nonbasic variables zero) is maximal.
Example continued – can this be further improved?

\[
\begin{align*}
\text{max} \quad z &= 13 - w_1 - 3x_2 - w_3 \\
\text{s.t.} \quad x_1 &= 2 - 2w_1 - 2x_2 + w_3 \\
\quad w_2 &= 1 + 2w_1 + 5x_2 \\
\quad x_3 &= 1 + 3w_1 + x_2 - 2w_3
\end{align*}
\]

1. **NO!**
2. All coefficients in objective negative (or zero).
3. Trivial solution (all nonbasic variables zero) is maximal.
Example continued – can this be further improved?

\[
\begin{align*}
\text{max} & \quad z = 13 - w_1 - 3x_2 - w_3 \\
\text{s.t.} & \quad x_1 = 2 - 2w_1 - 2x_2 + w_3 \\
& \quad w_2 = 1 + 2w_1 + 5x_2 \\
& \quad x_3 = 1 + 3w_1 + x_2 - 2w_3
\end{align*}
\]

1. NO!
2. All coefficients in objective negative (or zero).
3. trivial solution (all nonbasic variables zero) is maximal.
Example continued – can this be further improved?

\[
\begin{align*}
\text{max} & \quad z = 13 - w_1 - 3x_2 - w_3 \\
\text{s.t.} & \quad x_1 = 2 - 2w_1 - 2x_2 + w_3 \\
& \quad w_2 = 1 + 2w_1 + 5x_2 \\
& \quad x_3 = 1 + 3w_1 + x_2 - 2w_3
\end{align*}
\]

1. NO!
2. All coefficients in objective negative (or zero).
3. Trivial solution (all nonbasic variables zero) is maximal.
Pivoting changes nothing

**Observation**

Every pivoting step just rewrites the LP into EQUIVALENT LP. When LP objective can no longer be improved because of rewrite, it implies that the original LP objective function can not be increased any further.
Simplex algorithm – summary

1. This was an informal description of the simplex algorithm.
2. At each step pivot on a nonbasic variable that improves objective function.
3. Till reach optimal solution.
4. Problem: Assumed that the starting (trivial) solution (all zero nonbasic vars) is feasible.
Starting somewhere...

\[ \max z = v + \sum_{j \in N} c_j x_j, \]

s.t. \[ x_i = b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for } i \in B, \]

\[ x_i \geq 0, \quad \forall i = 1, \ldots, n + m. \]

1. \( L \): Transformed LP to slack form.

2. Simplex starts from feasible solution and walks around till reaches opt.

3. \( L \) might not be feasible at all.

4. Example on left, trivial sol is not feasible, if \( \exists b_i < 0 \).

Idea: Add a variable \( x_0 \), and minimize it!

\[ \min x_0 \]

s.t. \[ x_i = x_0 + b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for } i \in B, \]

\[ x_i \geq 0, \quad \forall i = 1, \ldots, n + m. \]
Starting somewhere...

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

1. **L**: Transformed LP to slack form.
2. **Simplex** starts from feasible solution and walks around till reaches opt.
3. \(L\) might not be feasible at all.
4. Example on left, trivial sol is not feasible, if \(\exists b_i < 0\).

Idea: Add a variable \(x_0\), and minimize it!

\[
\begin{align*}
\text{min} & \quad x_0 \\
\text{s.t.} & \quad x_i = x_0 + b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for } i \in B, \\
& \quad x_i \geq 0, \quad \forall i = 1, \ldots, n + m.
\end{align*}
\]
Starting somewhere...

\[ \max z = v + \sum_{j \in N} c_j x_j, \]
\[ \text{s.t. } x_i = b_i - \sum_{j \in N} a_{ij} x_j \text{ for } i \in B, \]
\[ x_i \geq 0, \quad \forall i = 1, \ldots, n + m. \]

1. **L**: Transformed \( \text{LP} \) to slack form.
2. **Simplex** starts from feasible solution and walks around till reaches opt.
3. \( L \) might not be feasible at all.
4. Example on left, trivial sol is not feasible, if \( \exists b_i < 0 \).

Idea: Add a variable \( x_0 \), and minimize it!

\[ \min x_0 \]
\[ \text{s.t. } x_i = x_0 + b_i - \sum_{j \in N} a_{ij} x_j \text{ for } i \in B, \]
\[ x_i \geq 0, \quad \forall i = 1, \ldots, n + m. \]
Starting somewhere...

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

1. **L**: Transformed LP to slack form.
2. **Simplex** starts from feasible solution and walks around till reaches opt.
3. **L** might not be feasible at all.
4. Example on left, trivial sol is not feasible, if \( \exists b_i < 0 \).

Idea: Add a variable \( x_0 \), and minimize it!

\[
\begin{align*}
\text{min} \quad x_0 \\
\text{s.t.} \quad x_i &= x_0 + b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for} \ i \in B, \\
&\quad x_i \geq 0, \quad \forall i = 1, \ldots, n + m.
\end{align*}
\]
Starting somewhere...

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

1. **L**: Transformed LP to slack form.
2. **Simplex** starts from feasible solution and walks around till reaches opt.
3. **L** might not be feasible at all.
4. Example on left, trivial sol is not feasible, if \( \exists b_i < 0 \).

Idea: Add a variable \( x_0 \), and minimize it!

\[ \begin{align*}
\text{min} \quad x_0 \\
\text{s.t.} \quad x_i &= x_0 + b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for } i \in B, \\
&\quad x_i \geq 0, \quad \forall i = 1, \ldots, n + m.
\end{align*} \]
Starting somewhere...

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

1. $L$: Transformed LP to slack form.
2. Simplex starts from feasible solution and walks around till reaches opt.
3. $L$ might not be feasible at all.
4. Example on left, trivial sol is not feasible, if $\exists b_i < 0$.

Idea: Add a variable $x_0$, and minimize it!

$$\begin{align*}
\text{min} \quad & x_0 \\
\text{s.t.} \quad & x_i = x_0 + b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for } i \in B, \\
& x_i \geq 0, \quad \forall i = 1, \ldots, n + m.
\end{align*}$$
Finding a feasible solution...

1. $L' = \text{Feasible}(L)$ (see previous slide).
2. Add new variable $x_0$ and make it large enough.
3. $x_0 = \max(-\min_i b_i, 0), \forall i > 0, x_i = 0$: feasible!
4. $\text{LPStartSolution}(L')$: Solution of Simplex to $L'$.
5. If $x_0$ is solution then $L$ feasible, and we found a valid basic solution.
6. If $x_0 > 0$ then LP not feasible. Done.
Finding a feasible solution...

1. $L' = \text{Feasible}(L)$ (see previous slide).
2. Add new variable $x_0$ and make it large enough.
3. $x_0 = \max(-\min_i b_i, 0), \forall i > 0, x_i = 0$: feasible!
4. $\text{LPStartSolution}(L')$: Solution of Simplex to $L'$.
5. If $x_0$ is solution then $L$ feasible, and we found a valid basic solution.
6. If $x_0 > 0$ then LP not feasible. Done.
4. \( L' = \text{Feasible}(L) \) (see previous slide).
2. Add new variable \( x_0 \) and make it large enough.
3. \( x_0 = \max(-\min_i b_i, 0), \forall i > 0, x_i = 0 \): feasible!
4. \text{LPStartSolution}(L') \): Solution of \text{Simplex} to \( L' \).
5. If \( x_0 \) is solution then \( L \) feasible, and we found a valid basic solution.
6. If \( x_0 > 0 \) then \( \text{LP} \) not feasible. Done.
Finding a feasible solution...

1. $L' = \text{Feasible}(L)$ (see previous slide).
2. Add new variable $x_0$ and make it large enough.
3. $x_0 = \max(-\min_i b_i, 0), \forall i > 0, x_i = 0$: feasible!
4. \text{LPStartSolution}(L')$: Solution of \text{Simplex} to $L'$.
5. If $x_0$ is solution then $L$ feasible, and we found a valid basic solution.
6. If $x_0 > 0$ then LP not feasible. Done.
Finding a feasible solution...

1. $L' = \text{Feasible}(L)$ (see previous slide).
2. Add new variable $x_0$ and make it large enough.
3. $x_0 = \max(-\min_i b_i, 0), \forall i > 0, x_i = 0$: feasible!
4. $\text{LPStartSolution}(L')$: Solution of Simplex to $L'$.
5. If $x_0$ is solution then $L$ feasible, and we found a valid basic solution.
6. If $x_0 > 0$ then LP not feasible. Done.
Lemma

\[ \text{LP } L \text{ is feasible } \iff \text{optimal objective value of LP } L' \text{ is zero.} \]

Proof.

A feasible solution to \( L \) is immediately an optimal solution to \( L' \) with \( x_0 = 0 \), and vice versa. Namely, given a solution to \( L' \) with \( x_0 = 0 \) we can transform it to a feasible solution to \( L \) by removing \( x_0 \).
Technicalities, technicalities everywhere

1. Starting solution for $L'$, generated by \texttt{LPStartSolution}(L).
2. .. not legal in slack form as non-basic variable $x_0$ assigned non-zero value.
3. Trick: Immediately pivoting on $x_0$ when running \texttt{Simplex}(L').
4. First try to decrease $x_0$ as much as possible.
Technicalities, technicalities everywhere

1. Starting solution for $L'$, generated by $\text{LPStartSolution}(L)$.

2. .. not legal in slack form as non-basic variable $x_0$ assigned non-zero value.

3. Trick: Immediately pivoting on $x_0$ when running $\text{Simplex}(L')$.

4. First try to decrease $x_0$ as much as possible.
Technicalities, technicalities everywhere

1. Starting solution for $L'$, generated by $\text{LPStartSolution}(L)$.
2. .. not legal in slack form as non-basic variable $x_0$ assigned non-zero value.
3. Trick: Immediately pivoting on $x_0$ when running $\text{Simplex}(L')$.
4. First try to decrease $x_0$ as much as possible.
1. Starting solution for $L'$, generated by $\text{LPStartSolution}(L)$.
2. .. not legal in slack form as non-basic variable $x_0$ assigned non-zero value.
3. Trick: Immediately pivoting on $x_0$ when running $\text{Simplex}(L')$.
4. First try to decrease $x_0$ as much as possible.