

# Linear Programming

Lecture 18

October 27, 2015

# 18.1: Linear Programming

# 18.1.1: Introduction and Motivation

# 18.1.1.1: Resource allocation in a factory

# A Factory Example

## Problem

Suppose a factory produces two products  $I$  and  $II$ . Each requires three resources  $A, B, C$ .

- 1 Producing one unit of Product I requires 1 unit each of resources  $A$  and  $C$ .
- 2 One unit of Product II requires 1 unit of resource  $B$  and 1 units of resource  $C$ .
- 3 We have 200 units of  $A$ , 300 units of  $B$ , and 400 units of  $C$ .
- 4 Product I can be sold for  $\$1$  and product II for  $\$6$ .

How many units of product I and product II should the factory manufacture to maximize profit?

**Solution:** Formulate as a linear program.

# A Factory Example

## Problem

Suppose a factory produces two products  $I$  and  $II$ . Each requires three resources  $A, B, C$ .

- 1 Producing one unit of Product I requires 1 unit each of resources  $A$  and  $C$ .
- 2 One unit of Product II requires 1 unit of resource  $B$  and 1 units of resource  $C$ .
- 3 We have 200 units of  $A$ , 300 units of  $B$ , and 400 units of  $C$ .
- 4 Product I can be sold for  $\$1$  and product II for  $\$6$ .

How many units of product I and product II should the factory manufacture to maximize profit?

**Solution:** Formulate as a linear program.

# A Factory Example

## Problem

Suppose a factory produces two products  $I$  and  $II$ . Each requires three resources  $A, B, C$ .

- 1 Producing one unit of Product I requires 1 unit each of resources  $A$  and  $C$ .
- 2 One unit of Product II requires 1 unit of resource  $B$  and 1 units of resource  $C$ .
- 3 We have 200 units of  $A$ , 300 units of  $B$ , and 400 units of  $C$ .
- 4 Product I can be sold for  $\$1$  and product II for  $\$6$ .

How many units of product I and product II should the factory manufacture to maximize profit?

**Solution:** Formulate as a linear program.

# A Factory Example

## Problem

Suppose a factory produces two products *I* and *II*. Each requires three resources *A*, *B*, *C*.

- 1 Producing unit I: Req. 1 unit of *A*, *C*.
- 2 Producing unit II: Requ. 1 unit of *B*, *C*.
- 3 Have *A*: 200, *B*: 300, and *C*: 400.
- 4 Price I: \$1, and II: \$6.

How many units of I and II to manufacture to max profit?



# A Factory Example

## Problem

Suppose a factory produces two products *I* and *II*. Each requires three resources *A*, *B*, *C*.

- 1 Producing unit I: Req. 1 unit of *A*, *C*.
- 2 Producing unit II: Requ. 1 unit of *B*, *C*.
- 3 Have *A*: 200, *B*: 300, and *C*: 400.
- 4 Price I: \$1, and II: \$6.

How many units of I and II to manufacture to max profit?

$$\begin{aligned} \max \quad & x_I + 6x_{II} \\ \text{s.t.} \quad & x_I \leq 200 & (A) \\ & x_{II} \leq 300 & (B) \\ & x_I + x_{II} \leq 400 & (C) \\ & x_I \geq 0 \\ & x_{II} \geq 0 \end{aligned}$$

# Linear Programming Formulation

Let us produce  $x_1$  units of product I and  $x_2$  units of product II. Our profit can be computed by solving

$$\begin{aligned} &\text{maximize} && x_1 + 6x_2 \\ &\text{subject to} && x_1 \leq 200 \\ & && x_2 \leq 300 \\ & && x_1 + x_2 \leq 400 \\ & && x_1, x_2 \geq 0 \end{aligned}$$

What is the solution?

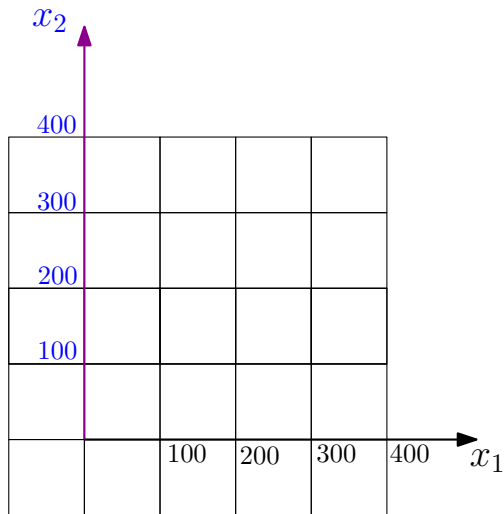
# Linear Programming Formulation

Let us produce  $x_1$  units of product I and  $x_2$  units of product II. Our profit can be computed by solving

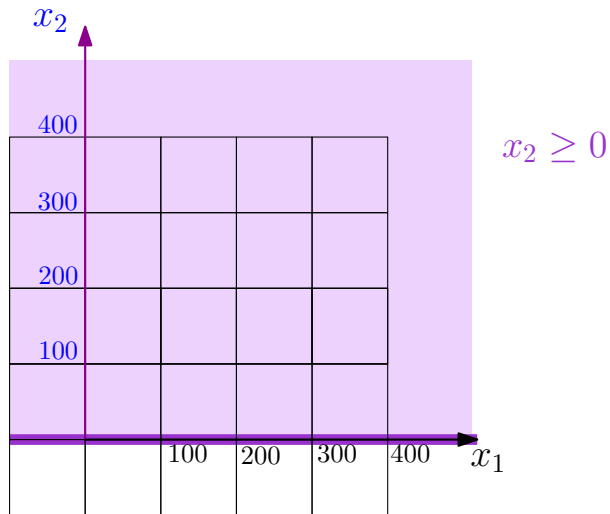
$$\begin{aligned} &\text{maximize} && x_1 + 6x_2 \\ &\text{subject to} && x_1 \leq 200 \\ & && x_2 \leq 300 \\ & && x_1 + x_2 \leq 400 \\ & && x_1, x_2 \geq 0 \end{aligned}$$

What is the solution?

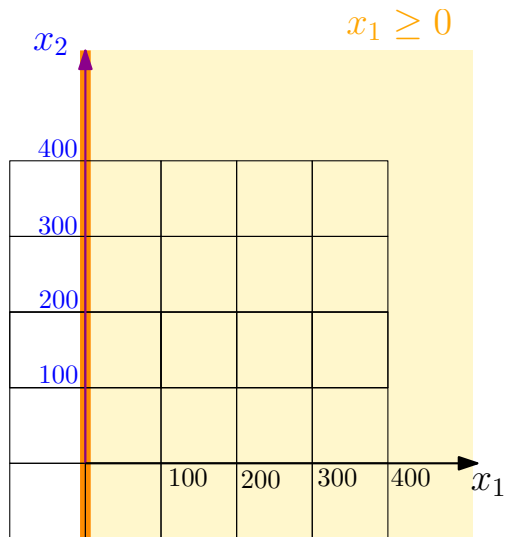
# Graphical interpretation of LP



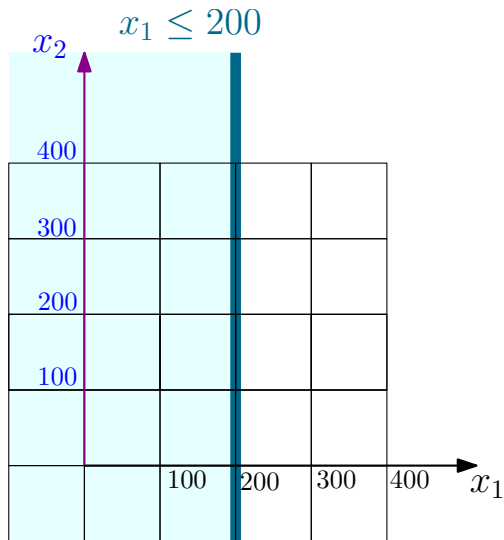
# Graphical interpretation of LP



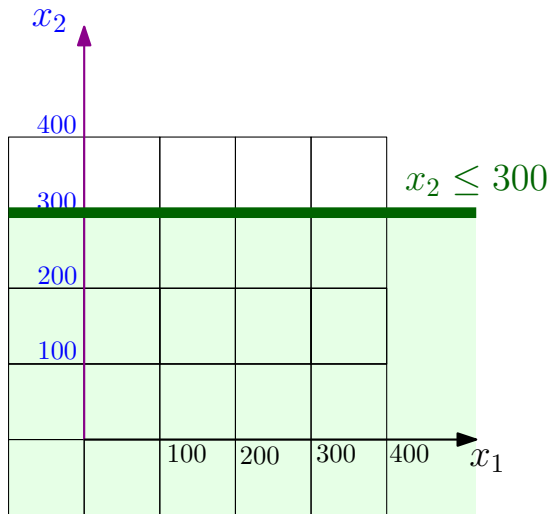
# Graphical interpretation of LP



# Graphical interpretation of LP

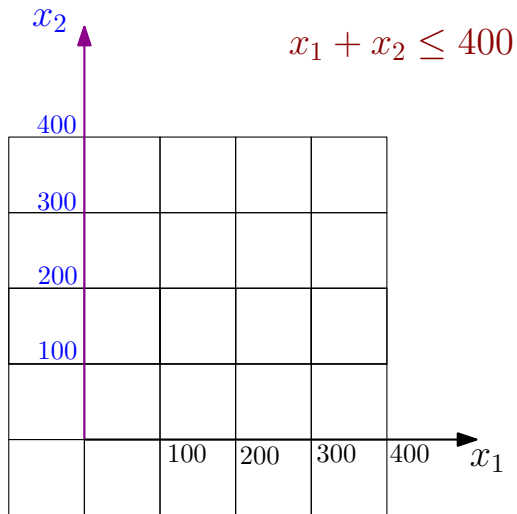


# Graphical interpretation of LP

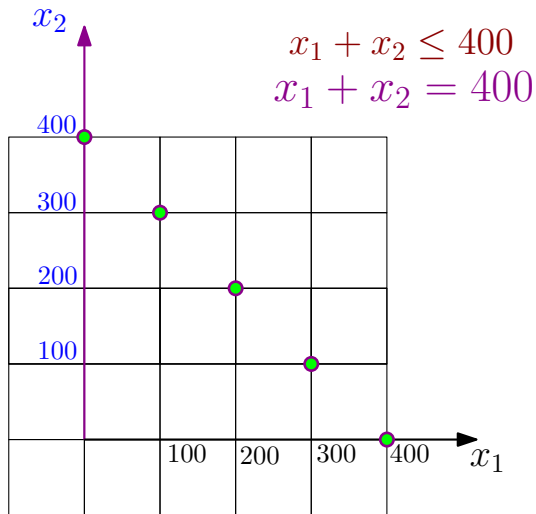




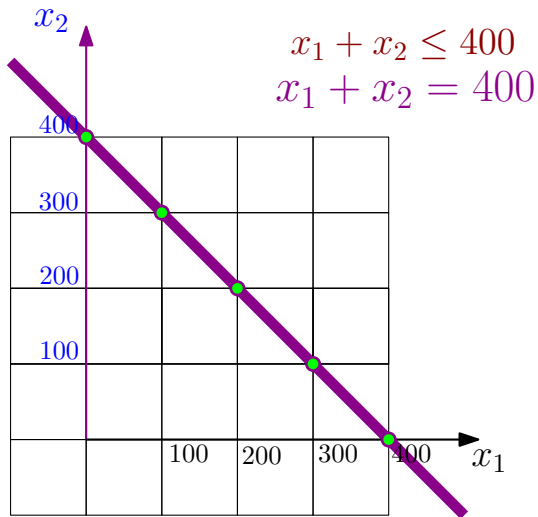
# Graphical interpretation of LP



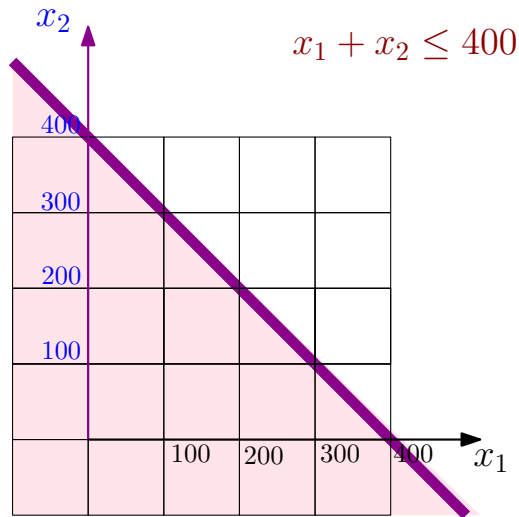
# Graphical interpretation of LP



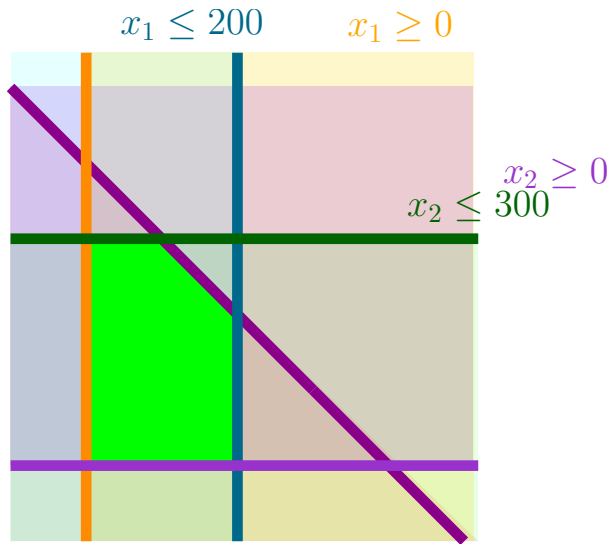
# Graphical interpretation of LP



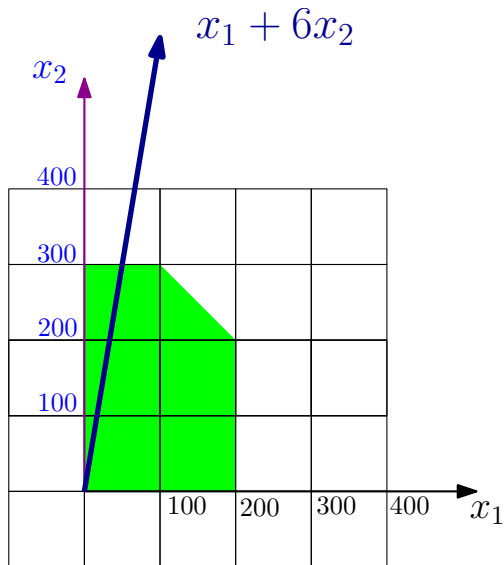
# Graphical interpretation of LP



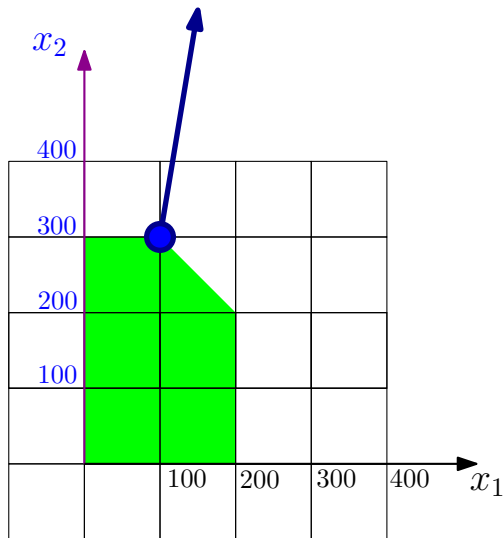
# Graphical interpretation of LP



# Graphical interpretation of LP



# Graphical interpretation of LP



## 18.1.1.2: More examples...



# Economic planning

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

- 1 Penguinia: 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 Penguinia need to prove a certain level of security:

$$x_{gun} + 1000 * x_{nuclear-bomb} \geq 1000,$$

where  $x_{guns}$ : # guns  $x_{nuclear-bomb}$ : # nuclear-bombs constructed.

- 6  $100 * x_{gun} + 1000000 * x_{nuclear-bomb} \leq x_{security}$   
 $x_{security}$ : total amount spent on security.

100/1, 000, 000: price of producing a single gun/nuclear bomb.

# Linear programming

An instance of **linear programming** (LP):

- ①  $x_1, \dots, x_n$ : variables.
- ② For  $j = 1, \dots, m$ :  $a_{j1}x_1 + \dots + a_{jn}x_n \leq b_j$ : linear inequality.
- ③ i.e., **constraint**.
- ④ Q:  $\exists$  assignment of values to  $x_1, \dots, x_n$  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

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n &\leq b_2 \\ \dots & \\ a_{m1}x_1 + \dots + a_{mn}x_n &\leq b_m \\ \max \quad c_1x_1 + \dots + c_nx_n. & \end{aligned}$$

# Linear Programming: A History

- 1 First formalized applied to problems in economics by Leonid Kantorovich in the 1930s
  - 1 However, work was ignored behind the Iron Curtain and unknown in the West
- 2 Rediscovered by Tjalling Koopmans in the 1940s, along with applications to economics
- 3 First algorithm (Simplex) to solve linear programs by George Dantzig in 1947
- 4 Kantorovich and Koopmans receive Nobel Prize for economics in 1975 ; Dantzig, however, was ignored
  - 1 Koopmans contemplated refusing the Nobel Prize to protest Dantzig's exclusion, but Kantorovich saw it as a vindication for using mathematics in economics, which had been written off as "a means for apologists of capitalism"

# Linear Programming: A History

- 1 First formalized applied to problems in economics by Leonid Kantorovich in the 1930s
  - 1 However, work was ignored behind the Iron Curtain and unknown in the West
- 2 Rediscovered by Tjalling Koopmans in the 1940s, along with applications to economics
- 3 First algorithm (Simplex) to solve linear programs by George Dantzig in 1947
- 4 Kantorovich and Koopmans receive Nobel Prize for economics in 1975 ; Dantzig, however, was ignored
  - 1 Koopmans contemplated refusing the Nobel Prize to protest Dantzig's exclusion, but Kantorovich saw it as a vindication for using mathematics in economics, which had been written off as "a means for apologists of capitalism"

# Linear Programming: A History

- 1 First formalized applied to problems in economics by Leonid Kantorovich in the 1930s
  - 1 However, work was ignored behind the Iron Curtain and unknown in the West
- 2 Rediscovered by Tjalling Koopmans in the 1940s, along with applications to economics
- 3 First algorithm (Simplex) to solve linear programs by George Dantzig in 1947
- 4 Kantorovich and Koopmans receive Nobel Prize for economics in 1975 ; Dantzig, however, was ignored
  - 1 Koopmans contemplated refusing the Nobel Prize to protest Dantzig's exclusion, but Kantorovich saw it as a vindication for using mathematics in economics, which had been written off as "a means for apologists of capitalism"

# Linear Programming: A History

- 1 First formalized applied to problems in economics by Leonid Kantorovich in the 1930s
  - 1 However, work was ignored behind the Iron Curtain and unknown in the West
- 2 Rediscovered by Tjalling Koopmans in the 1940s, along with applications to economics
- 3 First algorithm (Simplex) to solve linear programs by George Dantzig in 1947
- 4 Kantorovich and Koopmans receive Nobel Prize for economics in 1975 ; Dantzig, however, was ignored
  - 1 Koopmans contemplated refusing the Nobel Prize to protest Dantzig's exclusion, but Kantorovich saw it as a vindication for using mathematics in economics, which had been written off as "a means for apologists of capitalism"

# Linear Programming: A History

- 1 First formalized applied to problems in economics by Leonid Kantorovich in the 1930s
  - 1 However, work was ignored behind the Iron Curtain and unknown in the West
- 2 Rediscovered by Tjalling Koopmans in the 1940s, along with applications to economics
- 3 First algorithm (Simplex) to solve linear programs by George Dantzig in 1947
- 4 Kantorovich and Koopmans receive Nobel Prize for economics in 1975 ; Dantzig, however, was ignored
  - 1 Koopmans contemplated refusing the Nobel Prize to protest Dantzig's exclusion, but Kantorovich saw it as a vindication for using mathematics in economics, which had been written off as "a means for apologists of capitalism"



# Linear Programming: A History

- 1 First formalized applied to problems in economics by Leonid Kantorovich in the 1930s
  - 1 However, work was ignored behind the Iron Curtain and unknown in the West
- 2 Rediscovered by Tjalling Koopmans in the 1940s, along with applications to economics
- 3 First algorithm (Simplex) to solve linear programs by George Dantzig in 1947
- 4 Kantorovich and Koopmans receive Nobel Prize for economics in 1975 ; Dantzig, however, was ignored
  - 1 Koopmans contemplated refusing the Nobel Prize to protest Dantzig's exclusion, but Kantorovich saw it as a vindication for using mathematics in economics, which had been written off as "a means for apologists of capitalism"

# Linear Programming: A History

- 1 First formalized applied to problems in economics by Leonid Kantorovich in the 1930s
  - 1 However, work was ignored behind the Iron Curtain and unknown in the West
- 2 Rediscovered by Tjalling Koopmans in the 1940s, along with applications to economics
- 3 First algorithm (Simplex) to solve linear programs by George Dantzig in 1947
- 4 Kantorovich and Koopmans receive Nobel Prize for economics in 1975 ; Dantzig, however, was ignored
  - 1 Koopmans contemplated refusing the Nobel Prize to protest Dantzig's exclusion, but Kantorovich saw it as a vindication for using mathematics in economics, which had been written off as "a means for apologists of capitalism"

# Network flow via linear programming

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

$$\forall (u, v) \in E \quad \begin{aligned} 0 &\leq x_{u \rightarrow v} \\ x_{u \rightarrow v} &\leq c(u \rightarrow v) \end{aligned}$$

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

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

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

# Maximum weight matching

Input:  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$  and weight  $w(\cdot)$  on the edges. Compute max matching in  $\mathbf{G}$ .

$$\forall uv \in \mathbf{E} \quad 0 \leq x_{uv} \\ x_{uv} \leq 1$$

$$\forall v \in \mathbf{V} \quad \sum_{uv \in \mathbf{E}} x_{uv} \leq 1$$

$$\max \quad \sum_{uv \in \mathbf{E}} w(uv)x_{uv}$$

## 18.1.1.3: Shortest path as a LP

# 18.2: The Simplex Algorithm

## 18.2.1: Linear program where all the variables are positive

# Rewriting an LP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{subject to} & \sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \dots, m \end{array}$$

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



# Rewriting an LP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{subject to} & \sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \dots, m \end{array}$$

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

# Rewriting an LP

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{subject to} & \sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \dots, m \end{array}$$

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

# 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**

## 18.2.2: Standard form

# Standard form of LP

## A linear program in standard form.

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j \\ \text{subject to} & \sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \dots, m \\ & x_j \geq 0 \quad \text{for } j = 1, \dots, n. \end{array}$$

# Standard form of LP

Because everything is clearer when you use matrices. Not.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1(n-1)} & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2(n-1)} & a_{2n} \\ \vdots & \dots & \dots & \dots & \vdots \\ a_{(m-1)1} & a_{(m-1)2} & \dots & a_{(m-1)(n-1)} & a_{(m-1)n} \\ a_{m1} & a_{m2} & \dots & a_{m(n-1)} & a_{mn} \end{pmatrix},$$

$c$ ,  $b$  and  $A$ :  
prespecified.

$x$  is vector of  
unknowns.

Solve **LP** for  
 $x$ .

## LP in standard form.

(Matrix notation.)

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b. \\ & x \geq 0. \end{aligned}$$

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix}.$$

## 18.2.3: Slack Form



# 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.

$$\begin{array}{ll} \max & c^T x \\ \text{subject to} & Ax = b. \\ & x \geq 0. \end{array}$$

- 1 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.

# 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.

$$\begin{array}{ll} \max & c^T x \\ \text{subject to} & Ax = b. \\ & x \geq 0. \end{array}$$

- 1 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.

# 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.

$$\begin{array}{ll} \max & c^T x \\ \text{subject to} & Ax = b. \\ & x \geq 0. \end{array}$$

- 1 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.

# 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.

$$\begin{array}{ll} \max & c^T x \\ \text{subject to} & Ax = b. \\ & x \geq 0. \end{array}$$

- 1 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.

# 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.

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

# 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.

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

# 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.

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

# 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.

$$\max \quad z = v + \sum_{j \in N} c_j x_j,$$

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

$$x_i \geq 0, \quad \forall i = 1, \dots, n + m.$$



# Basic/nonbasic

Nonbasic variables

$$\begin{aligned} \max \quad z &= 29 - \frac{1}{9}x_3 - \frac{1}{9}x_5 - \frac{2}{9}x_6 \\ x_1 &= 8 + \frac{1}{6}x_3 + \frac{1}{6}x_5 - \frac{1}{3}x_6 \\ x_2 &= 4 - \frac{8}{3}x_3 - \frac{2}{3}x_5 + \frac{1}{3}x_6 \\ x_4 &= 18 - \frac{1}{2}x_3 + \frac{1}{2}x_5 \end{aligned}$$

Basic variables

# 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

$N \cup B = \{1, \dots, n + m\}$

$v$  - objective function constant.

# Slack form formally

## Final form

$$\max \quad z = v + \sum_{j \in N} c_j x_j,$$

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

$$x_i \geq 0, \quad \forall i = 1, \dots, n + m.$$

# Example

Consider the following LP which is in slack form.

$$\max \quad z = 29 - \frac{1}{9}x_3 - \frac{1}{9}x_5 - \frac{2}{9}x_6$$

$$x_1 = 8 + \frac{1}{6}x_3 + \frac{1}{6}x_5 - \frac{1}{3}x_6$$

$$x_2 = 4 - \frac{8}{3}x_3 - \frac{2}{3}x_5 + \frac{1}{3}x_6$$

$$x_4 = 18 - \frac{1}{2}x_3 + \frac{1}{2}x_5$$

# Example

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

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

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

$$b = \begin{pmatrix} b_1 \\ b_2 \\ b_4 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 18 \end{pmatrix} \quad c = \begin{pmatrix} c_3 \\ c_5 \\ c_6 \end{pmatrix} = \begin{pmatrix} -1/9 \\ -1/9 \\ -2/9 \end{pmatrix}$$

$$v = 29.$$

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...

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

Transform into slack form...

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

## 18.2.4: The Simplex algorithm by example

# The Simplex algorithm by example

$$\begin{array}{ll}\max & 5x_1 + 4x_2 + 3x_3 \\ \text{s.t.} & 2x_1 + 3x_2 + x_3 \leq 5 \\ & 4x_1 + x_2 + 2x_3 \leq 11 \\ & 3x_1 + 4x_2 + 2x_3 \leq 8 \\ & x_1, x_2, x_3 \geq 0\end{array}$$

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

$$\begin{array}{ll}\max & z = 5x_1 + 4x_2 + 3x_3 \\ \text{s.t.} & w_1 = 5 - 2x_1 - 3x_2 - x_3 \\ \Rightarrow & w_2 = 11 - 4x_1 - x_2 - 2x_3 \\ & w_3 = 8 - 3x_1 - 4x_2 - 2x_3 \\ & x_1, x_2, x_3, w_1, w_2, w_3 \geq 0\end{array}$$



# Example continued I...

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

①  $\implies w_1 = 5, w_2 = 11$  and  $w_3 = 8$ .

② Feasible!

③ Objection function value:  $z = 0$ .

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

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:  
 $x_1 = x_2 = x_3 = 0$ .

# Example continued I...

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

①  $\implies w_1 = 5, w_2 = 11$  and  $w_3 = 8$ .

② Feasible!

③ Objection function value:  $z = 0$ .

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

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:  
 $x_1 = x_2 = x_3 = 0$ .

# Example continued I...

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

①  $\implies w_1 = 5, w_2 = 11$  and  $w_3 = 8$ .

② Feasible!

③ Objection function value:  $z = 0$ .

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

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:  
 $x_1 = x_2 = x_3 = 0$ .

# Example continued I...

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

①  $\implies w_1 = 5, w_2 = 11$  and  $w_3 = 8$ .

② Feasible!

③ Objection function value:  $z = 0$ .

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

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:  
 $x_1 = x_2 = x_3 = 0$ .

# Example continued I...

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

①  $\implies w_1 = 5, w_2 = 11$  and  $w_3 = 8$ .

② Feasible!

③ Objection function value:  $z = 0$ .

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

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:

$$x_1 = x_2 = x_3 = 0.$$

# Example continued I...

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

①  $\implies w_1 = 5, w_2 = 11$  and  $w_3 = 8$ .

② Feasible!

③ Objection function value:  $z = 0$ .

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

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:  
 $x_1 = x_2 = x_3 = 0$ .

# Example continued I...

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

①  $\implies w_1 = 5, w_2 = 11$  and  $w_3 = 8$ .

② Feasible!

③ Objective function value:  $z = 0$ .

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

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:  
 $x_1 = x_2 = x_3 = 0$ .

## Example continued II...

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

- ①  $x_1 = x_2 = x_3 = 0$   
 $\implies w_1 = 5,$   
 $w_2 = 11$  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  $x_1$ ???
- ③ Careful! Might break feasibility.
- ④ Increase  $x_1$  as much as possible without breaking feasibility!



## Example continued II...

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

- 1  $x_1 = x_2 = x_3 = 0$   
 $\implies w_1 = 5,$   
 $w_2 = 11$  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  $x_1$ ???
- 3 Careful! Might break feasibility.
- 4 Increase  $x_1$  as much as possible without breaking feasibility!

## Example continued II...

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

- 1  $x_1 = x_2 = x_3 = 0$   
 $\implies w_1 = 5,$   
 $w_2 = 11$  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  $x_1$ ???
- 3 Careful! Might break feasibility.
- 4 Increase  $x_1$  as much as possible without breaking feasibility!

## Example continued II...

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

- 1  $x_1 = x_2 = x_3 = 0$   
 $\implies w_1 = 5,$   
 $w_2 = 11$  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  $x_1$ ???
- 3 Careful! Might break feasibility.
- 4 Increase  $x_1$  as much as possible without breaking feasibility!

## Example continued II...

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

- 1  $x_1 = x_2 = x_3 = 0$   
 $\implies w_1 = 5,$   
 $w_2 = 11$  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  $x_1$ ???
- 3 Careful! Might break feasibility.
- 4 Increase  $x_1$  as much as possible without breaking feasibility!

## Example continued II...

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

- 1  $x_1 = x_2 = x_3 = 0$   
 $\implies w_1 = 5,$   
 $w_2 = 11$  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  $x_1$ ???
- 3 Careful! Might break feasibility.
- 4 Increase  $x_1$  as much as possible without breaking feasibility!

## Example continued II...

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

- 1  $x_1 = x_2 = x_3 = 0$   
 $\implies w_1 = 5,$   
 $w_2 = 11$  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  $x_1$ ???
- 3 Careful! Might break feasibility.
- 4 Increase  $x_1$  as much as possible without breaking feasibility!

# Example continued III...

Set  $x_2 = x_3 = 0$

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

$$\begin{aligned} w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\ &= 5 - 2x_1 \end{aligned}$$

$$\begin{aligned} w_2 &= 11 - 4x_1 - x_2 - 2x_3 \\ &= 11 - 4x_1 \end{aligned}$$

$$\begin{aligned} w_3 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ &= 8 - 3x_1. \end{aligned}$$

① Want to increase  $x_1$  as much as possible, as long as:

$$w_1 = 5 - 2x_1 \geq 0,$$

$$w_2 = 11 - 4x_1 \geq 0,$$

$$\text{and } w_3 = 8 - 3x_1 \geq 0.$$

# Example continued III...

Set  $x_2 = x_3 = 0$

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

$$\begin{aligned} w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\ &= 5 - 2x_1 \end{aligned}$$

$$\begin{aligned} w_2 &= 11 - 4x_1 - x_2 - 2x_3 \\ &= 11 - 4x_1 \end{aligned}$$

$$\begin{aligned} w_3 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ &= 8 - 3x_1. \end{aligned}$$

① Want to increase  $x_1$  as much as possible, as long as:

$$w_1 = 5 - 2x_1 \geq 0,$$

$$w_2 = 11 - 4x_1 \geq 0,$$

$$\text{and } w_3 = 8 - 3x_1 \geq 0.$$



# Example continued III...

Set  $x_2 = x_3 = 0$

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

$$\begin{aligned} w_1 &= 5 - 2x_1 - 3x_2 - x_3 \\ &= 5 - 2x_1 \end{aligned}$$

$$\begin{aligned} w_2 &= 11 - 4x_1 - x_2 - 2x_3 \\ &= 11 - 4x_1 \end{aligned}$$

$$\begin{aligned} w_3 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ &= 8 - 3x_1. \end{aligned}$$

① Want to increase  $x_1$  as much as possible, as long as:

$$w_1 = 5 - 2x_1 \geq 0,$$

$$w_2 = 11 - 4x_1 \geq 0,$$

$$\text{and } w_3 = 8 - 3x_1 \geq 0.$$

# Example continued IV...

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

① Constraints:

$$w_1 = 5 - 2x_1 \geq 0,$$

$$w_2 = 11 - 4x_1 \geq 0,$$

$$\text{and } w_3 = 8 - 3x_1 \geq 0.$$

②  $x_1 \leq 2.5,$

$x_1 \leq 11/4 = 2.75$  and

$x_1 \leq 8/3 = 2.66$

- 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.

# Example continued IV...

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

① Constraints:

$$w_1 = 5 - 2x_1 \geq 0,$$

$$w_2 = 11 - 4x_1 \geq 0,$$

$$\text{and } w_3 = 8 - 3x_1 \geq 0.$$

②  $x_1 \leq 2.5$ ,  
 $x_1 \leq 11/4 = 2.75$  and  
 $x_1 \leq 8/3 = 2.66$

- 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.

# Example continued IV...

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

① Constraints:

$$w_1 = 5 - 2x_1 \geq 0,$$

$$w_2 = 11 - 4x_1 \geq 0,$$

$$\text{and } w_3 = 8 - 3x_1 \geq 0.$$

②  $x_1 \leq 2.5,$

$x_1 \leq 11/4 = 2.75$  and

① Maximum we can increase  $x_1$  is **2.5**.  $x_1 \leq 8/3 = 2.66$

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.

# Example continued IV...

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

① Constraints:

$$w_1 = 5 - 2x_1 \geq 0,$$

$$w_2 = 11 - 4x_1 \geq 0,$$

$$\text{and } w_3 = 8 - 3x_1 \geq 0.$$

②  $x_1 \leq 2.5,$

$x_1 \leq 11/4 = 2.75$  and

① Maximum we can increase  $x_1$  is  $2.5$ .  $x_1 \leq 8/3 = 2.66$

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.

# Example continued IV...

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

① Constraints:

$$w_1 = 5 - 2x_1 \geq 0,$$

$$w_2 = 11 - 4x_1 \geq 0,$$

$$\text{and } w_3 = 8 - 3x_1 \geq 0.$$

②  $x_1 \leq 2.5,$

$x_1 \leq 11/4 = 2.75$  and

$x_1 \leq 8/3 = 2.66$

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.

# Example continued IV...

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

① Constraints:

$$w_1 = 5 - 2x_1 \geq 0,$$

$$w_2 = 11 - 4x_1 \geq 0,$$

$$\text{and } w_3 = 8 - 3x_1 \geq 0.$$

②  $x_1 \leq 2.5,$

$x_1 \leq 11/4 = 2.75$  and

① Maximum we can increase  $x_1$  is  $2.5$ .  $x_1 \leq 8/3 = 2.66$

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.

# Example continued V...

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

- 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 V...

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

- 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 V...

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

- 1  $x_1 = 2.5, x_2 = 0, x_3 = 0, w_1 = 0, w_2 = 1, w_3 = 0.5$
- 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 V...

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

- 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 VI...

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

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

- 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 VI...

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

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

- 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 VI...

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

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

- 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 VII...

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

② pivoted on  $x_1$ .

③ Continue pivoting till reach optimal solution.

$$\max \quad z = 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3$$

$$x_1 = 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3$$

$$w_2 = 1 + 2w_1 + 5x_2$$

$$w_3 = 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.$$

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 VII...

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

② pivoted on  $x_1$ .

③ Continue pivoting till reach optimal solution.

$$\max \quad z = 12.5 - 2.5w_1 - 3.5x_2 + 0.5x_3$$

$$x_1 = 2.5 - 0.5w_1 - 1.5x_2 - 0.5x_3$$

$$w_2 = 1 + 2w_1 + 5x_2$$

$$w_3 = 0.5 + 1.5w_1 + 0.5x_2 - 0.5x_3.$$

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 VII...

- 1 Rewriting step done is called **pivoting**.
- 2 pivoted on  $x_1$ .
- 3 Continue pivoting till reach optimal solution.

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

- 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 VII...

- 1 Rewriting step done is called **pivoting**.
- 2 pivoted on  $x_1$ .
- 3 Continue pivoting till reach optimal solution.

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

- 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 VII...

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

② pivoted on  $x_1$ .

③ Continue pivoting till reach optimal solution.

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

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 VII...

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

② pivoted on  $x_1$ .

③ Continue pivoting till reach optimal solution.

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

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 VII...

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

② pivoted on  $x_1$ .

③ Continue pivoting till reach optimal solution.

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

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 VIII...

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

- 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.

$$\begin{aligned}\max \quad z &= 13 - w_1 - 3x_2 - w_3 \\ s.t. \quad x_1 &= 2 - 2w_1 - 2x_2 + w_3 \\ w_2 &= 1 + 2w_1 + 5x_2 \\ x_3 &= 1 + 3w_1 + x_2 - 2w_3\end{aligned}$$

## Example continued VIII...

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

- 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.

$$\begin{aligned}\max \quad z &= 13 - w_1 - 3x_2 - w_3 \\ s.t. \quad x_1 &= 2 - 2w_1 - 2x_2 + w_3 \\ w_2 &= 1 + 2w_1 + 5x_2 \\ x_3 &= 1 + 3w_1 + x_2 - 2w_3\end{aligned}$$

# Example continued – can this be further improved?

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

- 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?

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

- 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?

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

- 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?

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

- 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.

# 18.2.4.1: Starting somewhere

# Starting somewhere...

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

- 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!

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

# Starting somewhere...

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

- 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!

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



# Starting somewhere...

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

- 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!

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

# Starting somewhere...

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

- 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!

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

# Starting somewhere...

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

- 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!

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

# Starting somewhere...

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

- 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!

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

# Finding a feasible solution...

- ①  $L' = \mathbf{Feasible}(L)$  (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  $\mathbf{LPStartSolution}(L')$ : Solution of **Simplex** to  $L'$ .
- ⑤ If  $x_0 = 0$  in solution then  $L$  feasible. Have valid basic solution.
- ⑥ If  $x_0 > 0$  then  $LP$  not feasible. Done.

# Finding a feasible solution...

- ①  $L' = \mathbf{Feasible}(L)$  (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  $\mathbf{LPStartSolution}(L')$ : Solution of  $\mathbf{Simplex}$  to  $L'$ .
- ⑤ If  $x_0 = 0$  in solution then  $L$  feasible. Have valid basic solution.
- ⑥ If  $x_0 > 0$  then  $LP$  not feasible. Done.

# Finding a feasible solution...

- ①  $L' = \mathbf{Feasible}(L)$  (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  $\mathbf{LPStartSolution}(L')$ : Solution of  $\mathbf{Simplex}$  to  $L'$ .
- ⑤ If  $x_0 = 0$  in solution then  $L$  feasible. Have 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  $x_0 = 0$  in solution then  $L$  feasible. Have valid basic solution.
- ⑥ If  $x_0 > 0$  then **LP** not feasible. Done.



# Finding a feasible solution...

- ①  $L' = \mathbf{Feasible}(L)$  (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  $\mathbf{LPStartSolution}(L')$ : Solution of  $\mathbf{Simplex}$  to  $L'$ .
- ⑤ If  $x_0 = 0$  in solution then  $L$  feasible. Have valid basic solution.
- ⑥ If  $x_0 > 0$  then  $\mathbf{LP}$  not feasible. Done.

# Lemma...

## Lemma

LP  $L$  is feasible  $\iff$  optimal objective value of LP  $L'$  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  $x_0$ .  $\square$

# Technicalities, technicalities everywhere

- ① Starting solution for  $L'$ , generated by **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 **Simplex**( $L'$ ).
- ④ First try to decrease  $x_0$  as much as possible.

# Technicalities, technicalities everywhere

- ① Starting solution for  $L'$ , generated by **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 **Simplex**( $L'$ ).
- ④ First try to decrease  $x_0$  as much as possible.

# Technicalities, technicalities everywhere

- ① Starting solution for  $L'$ , generated by **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 **Simplex**( $L'$ ).
- ④ First try to decrease  $x_0$  as much as possible.

# Technicalities, technicalities everywhere

- ① Starting solution for  $L'$ , generated by **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 **Simplex**( $L'$ ).
- ④ First try to decrease  $x_0$  as much as possible.









