

Randomised Algorithms

Lecture 6: Linear Programming: Introduction

Thomas Sauerwald (tms41@cam.ac.uk)

Lent 2024



UNIVERSITY OF
CAMBRIDGE

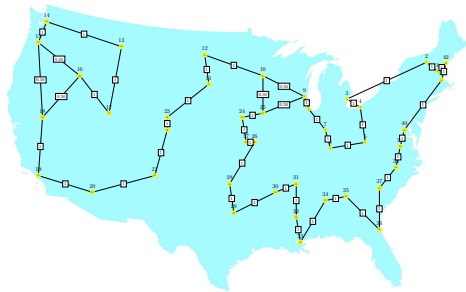
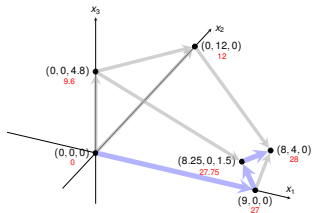
Outline

Introduction

A Simple Example of a Linear Program

Formulating Problems as Linear Programs

Standard and Slack Forms



- linear programming is a powerful tool in optimisation
- inspired more sophisticated techniques such as quadratic optimisation, convex optimisation, integer programming and semi-definite programming
- we will later use the connection between linear and integer programming to tackle several problems (Vertex-Cover, Set-Cover, TSP, satisfiability)

Outline

Introduction

A Simple Example of a Linear Program

Formulating Problems as Linear Programs

Standard and Slack Forms

What are Linear Programs?

Linear Programming (informal definition)

- maximise or minimise an objective, given limited resources (competing constraint)
- constraints are specified as (in)equalities
- objective function and constraints are **linear**

A Simple Example of a Linear Optimisation Problem

- Laptop

A Simple Example of a Linear Optimisation Problem

- Laptop
 - selling price to retailer: 1,000 GBP

A Simple Example of a Linear Optimisation Problem



- Laptop
 - selling price to retailer: 1,000 GBP
 - glass: 4 units

A Simple Example of a Linear Optimisation Problem



- Laptop
 - selling price to retailer: 1,000 GBP
 - glass: 4 units
 - copper: 2 units

A Simple Example of a Linear Optimisation Problem



■ Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit

A Simple Example of a Linear Optimisation Problem



- Laptop
 - selling price to retailer: 1,000 GBP
 - glass: 4 units
 - copper: 2 units
 - rare-earth elements: 1 unit

- Smartphone

A Simple Example of a Linear Optimisation Problem



- Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit

- Smartphone

- selling price to retailer: 1,000 GBP

A Simple Example of a Linear Optimisation Problem



■ Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit



■ Smartphone

- selling price to retailer: 1,000 GBP
- glass: 1 unit

A Simple Example of a Linear Optimisation Problem

■ Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit



■ Smartphone

- selling price to retailer: 1,000 GBP
- glass: 1 unit
- copper: 1 unit

A Simple Example of a Linear Optimisation Problem

■ Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit



■ Smartphone

- selling price to retailer: 1,000 GBP
- glass: 1 unit
- copper: 1 unit
- rare-earth elements: 2 units

A Simple Example of a Linear Optimisation Problem

■ Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit



■ Smartphone

- selling price to retailer: 1,000 GBP
- glass: 1 unit
- copper: 1 unit
- rare-earth elements: 2 units

- You have a **daily supply** of:

A Simple Example of a Linear Optimisation Problem

▪ Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit



▪ Smartphone

- selling price to retailer: 1,000 GBP
- glass: 1 unit
- copper: 1 unit
- rare-earth elements: 2 units



▪ You have a daily supply of:

- glass: 20 units

A Simple Example of a Linear Optimisation Problem

■ Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit



■ Smartphone

- selling price to retailer: 1,000 GBP
- glass: 1 unit
- copper: 1 unit
- rare-earth elements: 2 units



■ You have a daily supply of:

- glass: 20 units
- copper: 10 units



A Simple Example of a Linear Optimisation Problem

■ Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit



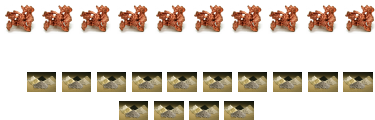
■ Smartphone

- selling price to retailer: 1,000 GBP
- glass: 1 unit
- copper: 1 unit
- rare-earth elements: 2 units



■ You have a daily supply of:

- glass: 20 units
- copper: 10 units
- rare-earth elements: 14 units



A Simple Example of a Linear Optimisation Problem

■ Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit



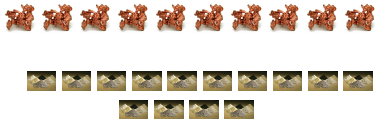
■ Smartphone

- selling price to retailer: 1,000 GBP
- glass: 1 unit
- copper: 1 unit
- rare-earth elements: 2 units



■ You have a daily supply of:

- glass: 20 units
- copper: 10 units
- rare-earth elements: 14 units
- (and enough of everything else...)



A Simple Example of a Linear Optimisation Problem

▪ Laptop

- selling price to retailer: 1,000 GBP
- glass: 4 units
- copper: 2 units
- rare-earth elements: 1 unit



▪ Smartphone

- selling price to retailer: 1,000 GBP
- glass: 1 unit
- copper: 1 unit
- rare-earth elements: 2 units



▪ You have a daily supply of:

- glass: 20 units
- copper: 10 units
- rare-earth elements: 14 units
- (and enough of everything else...)



How to maximise your daily earnings?

The Linear Program

Linear Program for the Production Problem

$$\begin{array}{llllll} \text{maximise} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & + & x_2 & \leq & 20 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & x_1 & + & 2x_2 & \leq & 14 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

The Linear Program

Linear Program for the Production Problem

$$\begin{array}{llllll} \text{maximise} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & + & x_2 & \leq & 20 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & x_1 & + & 2x_2 & \leq & 14 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

The solution of this linear program yields the optimal production schedule.

The Linear Program

Linear Program for the Production Problem

$$\begin{array}{llllll} \text{maximise} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & + & x_2 & \leq & 20 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & x_1 & + & 2x_2 & \leq & 14 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

The solution of this linear program yields the optimal production schedule.

Formal Definition of Linear Program

The Linear Program

Linear Program for the Production Problem

$$\begin{array}{llllll} \text{maximise} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & + & x_2 & \leq & 20 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & x_1 & + & 2x_2 & \leq & 14 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

The solution of this linear program yields the optimal production schedule.

Formal Definition of Linear Program

- Given a_1, a_2, \dots, a_n and a set of variables x_1, x_2, \dots, x_n , a **linear function** f is defined by

$$f(x_1, x_2, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n.$$

The Linear Program

Linear Program for the Production Problem

$$\begin{array}{llllll} \text{maximise} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & + & x_2 & \leq & 20 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & x_1 & + & 2x_2 & \leq & 14 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

The solution of this linear program yields the optimal production schedule.

Formal Definition of Linear Program

- Given a_1, a_2, \dots, a_n and a set of variables x_1, x_2, \dots, x_n , a **linear function** f is defined by

$$f(x_1, x_2, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n.$$

- Linear Equality:** $f(x_1, x_2, \dots, x_n) = b$
- Linear Inequality:** $f(x_1, x_2, \dots, x_n) \begin{matrix} \geq \\ \leq \end{matrix} b$

The Linear Program

Linear Program for the Production Problem

$$\begin{array}{llllll} \text{maximise} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & + & x_2 & \leq & 20 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & x_1 & + & 2x_2 & \leq & 14 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

The solution of this linear program yields the optimal production schedule.

Formal Definition of Linear Program

- Given a_1, a_2, \dots, a_n and a set of variables x_1, x_2, \dots, x_n , a **linear function** f is defined by

$$f(x_1, x_2, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n.$$

- Linear Equality:** $f(x_1, x_2, \dots, x_n) = b$
- Linear Inequality:** $f(x_1, x_2, \dots, x_n) \begin{matrix} \geq \\ \leq \end{matrix} b$

Linear Constraints

The Linear Program

Linear Program for the Production Problem

$$\begin{array}{llllll} \text{maximise} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & + & x_2 & \leq & 20 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & x_1 & + & 2x_2 & \leq & 14 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

The solution of this linear program yields the optimal production schedule.

Formal Definition of Linear Program

- Given a_1, a_2, \dots, a_n and a set of variables x_1, x_2, \dots, x_n , a **linear function** f is defined by

$$f(x_1, x_2, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n.$$

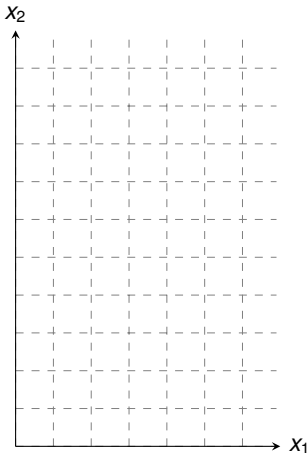
- Linear Equality:** $f(x_1, x_2, \dots, x_n) = b$
- Linear Inequality:** $f(x_1, x_2, \dots, x_n) \begin{matrix} \geq \\ \leq \end{matrix} b$
- Linear-Programming Problem:** either minimise or maximise a linear function subject to a set of linear constraints

Linear Constraints

Finding the Optimal Production Schedule

$$\begin{array}{rllll} \text{maximise} & x_1 & + & x_2 & \\ \text{subject to} & & & & \\ & 4x_1 & + & x_2 & \leq 20 \\ & 2x_1 & + & x_2 & \leq 10 \\ & x_1 & + & 2x_2 & \leq 14 \\ & x_1, x_2 & & & \geq 0 \end{array}$$

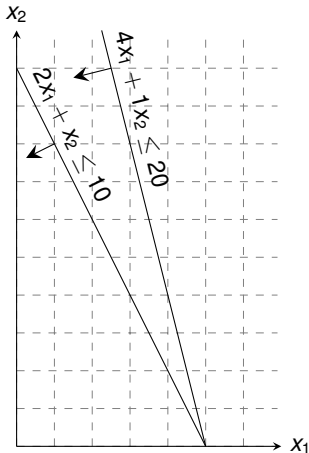
Any setting of x_1 and x_2 satisfying all constraints is a feasible solution



Finding the Optimal Production Schedule

$$\begin{array}{llllll} \text{maximise} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & + & x_2 & \leq & 20 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & x_1 & + & 2x_2 & \leq & 14 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

Any setting of x_1 and x_2 satisfying all constraints is a feasible solution

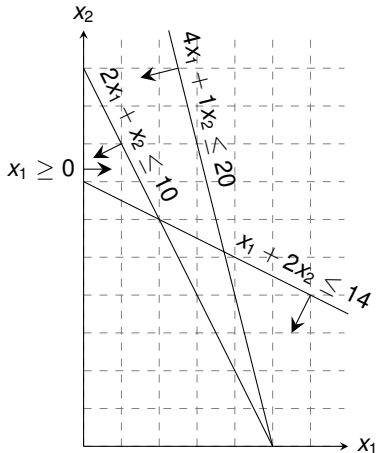


Finding the Optimal Production Schedule

maximise $x_1 + x_2$
subject to

$$\begin{array}{rclcl} 4x_1 & + & x_2 & \leq & 20 \\ 2x_1 & + & x_2 & \leq & 10 \\ x_1 & + & 2x_2 & \leq & 14 \\ x_1, x_2 & & & \geq & 0 \end{array}$$

Any setting of x_1 and x_2 satisfying all constraints is a feasible solution

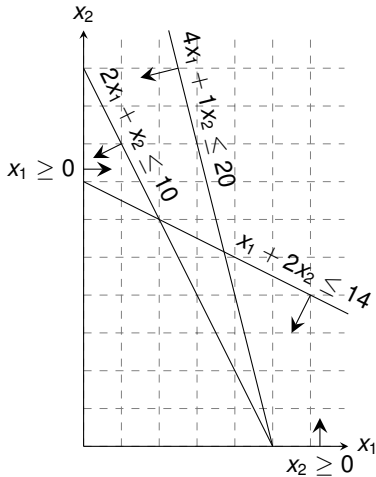


Finding the Optimal Production Schedule

maximise $x_1 + x_2$
subject to

$$\begin{array}{rclcl} 4x_1 & + & x_2 & \leq & 20 \\ 2x_1 & + & x_2 & \leq & 10 \\ x_1 & + & 2x_2 & \leq & 14 \\ x_1, x_2 & & & \geq & 0 \end{array}$$

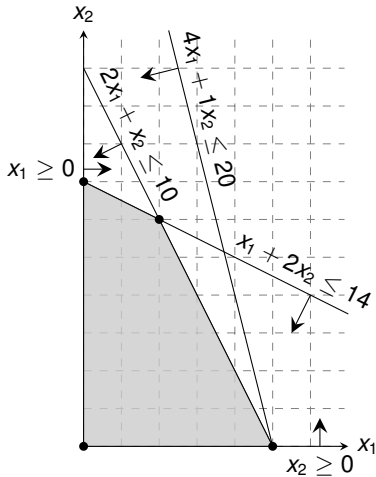
Any setting of x_1 and x_2 satisfying all constraints is a feasible solution



Finding the Optimal Production Schedule

$$\begin{array}{llllll} \text{maximise} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & + & x_2 & \leq & 20 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & x_1 & + & 2x_2 & \leq & 14 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

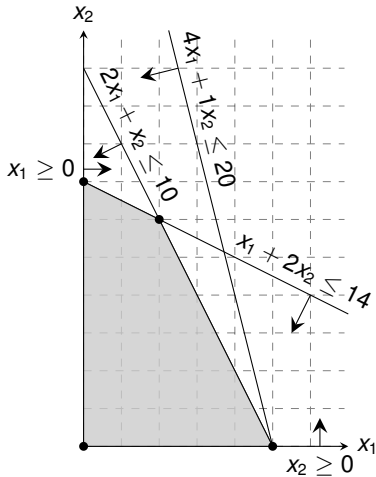
Any setting of x_1 and x_2 satisfying all constraints is a feasible solution



Finding the Optimal Production Schedule

$$\begin{array}{llll} \text{maximise} & x_1 & + & x_2 \\ \text{subject to} & & & \\ & 4x_1 & + & x_2 \leq 20 \\ & 2x_1 & + & x_2 \leq 10 \\ & x_1 & + & 2x_2 \leq 14 \\ & x_1, x_2 & & \geq 0 \end{array}$$

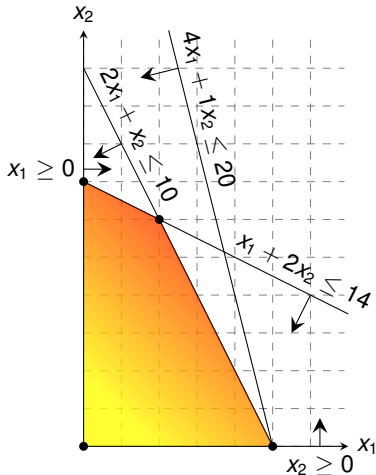
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far as possible.



Finding the Optimal Production Schedule

$$\begin{array}{llll} \text{maximise} & x_1 & + & x_2 \\ \text{subject to} & & & \\ & 4x_1 & + & x_2 \leq 20 \\ & 2x_1 & + & x_2 \leq 10 \\ & x_1 & + & 2x_2 \leq 14 \\ & x_1, x_2 & & \geq 0 \end{array}$$

Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.

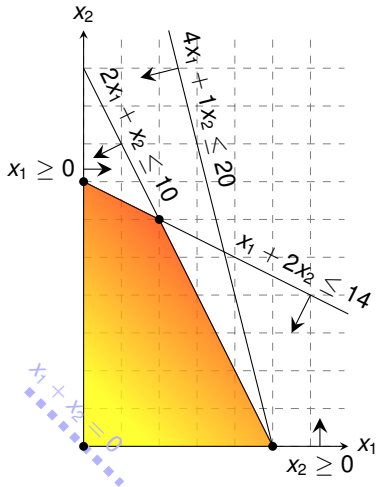


Finding the Optimal Production Schedule

maximise $x_1 + x_2$
subject to

$$\begin{array}{rclcl} 4x_1 & + & x_2 & \leq & 20 \\ 2x_1 & + & x_2 & \leq & 10 \\ x_1 & + & 2x_2 & \leq & 14 \\ x_1, x_2 & & & \geq & 0 \end{array}$$

Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.

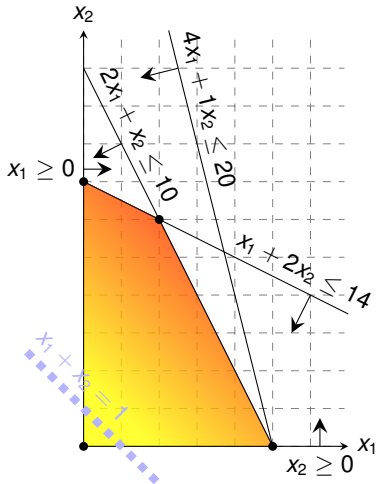


Finding the Optimal Production Schedule

maximise $x_1 + x_2$
subject to

$$\begin{array}{rclcl} 4x_1 & + & x_2 & \leq & 20 \\ 2x_1 & + & x_2 & \leq & 10 \\ x_1 & + & 2x_2 & \leq & 14 \\ x_1, x_2 & & & \geq & 0 \end{array}$$

Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.

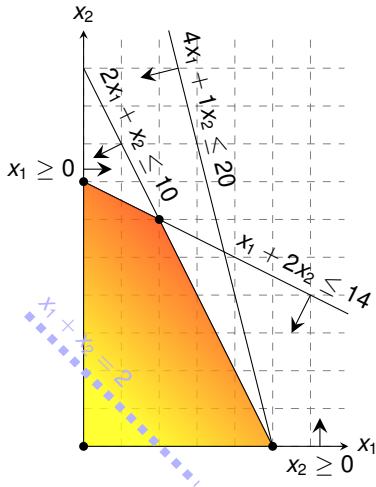


Finding the Optimal Production Schedule

maximise $x_1 + x_2$
subject to

$$\begin{array}{rclcl} 4x_1 & + & x_2 & \leq & 20 \\ 2x_1 & + & x_2 & \leq & 10 \\ x_1 & + & 2x_2 & \leq & 14 \\ x_1, x_2 & & & \geq & 0 \end{array}$$

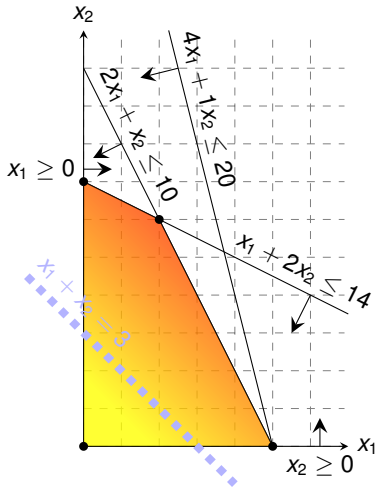
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far as possible.



Finding the Optimal Production Schedule

$$\begin{array}{llll} \text{maximise} & x_1 & + & x_2 \\ \text{subject to} & & & \\ & 4x_1 & + & x_2 \leq 20 \\ & 2x_1 & + & x_2 \leq 10 \\ & x_1 & + & 2x_2 \leq 14 \\ & x_1, x_2 & & \geq 0 \end{array}$$

Graphical Procedure: Move the line $x_1 + x_2 = z$ as far as possible.

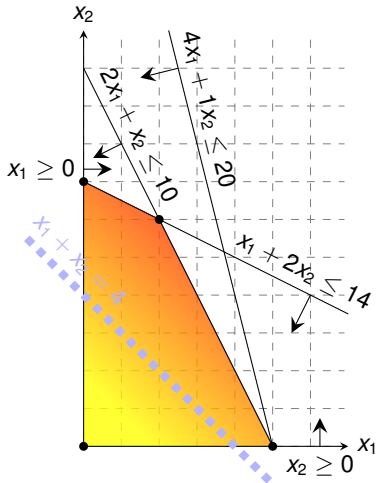


Finding the Optimal Production Schedule

maximise $x_1 + x_2$
subject to

$$\begin{array}{rclcl} 4x_1 & + & x_2 & \leq & 20 \\ 2x_1 & + & x_2 & \leq & 10 \\ x_1 & + & 2x_2 & \leq & 14 \\ x_1, x_2 & & & \geq & 0 \end{array}$$

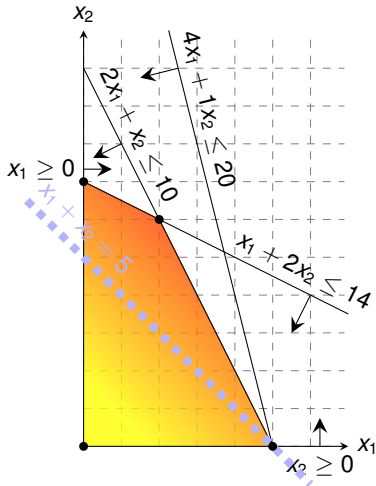
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far as possible.



Finding the Optimal Production Schedule

$$\begin{array}{llll} \text{maximise} & x_1 & + & x_2 \\ \text{subject to} & & & \\ & 4x_1 & + & x_2 \leq 20 \\ & 2x_1 & + & x_2 \leq 10 \\ & x_1 & + & 2x_2 \leq 14 \\ & x_1, x_2 & & \geq 0 \end{array}$$

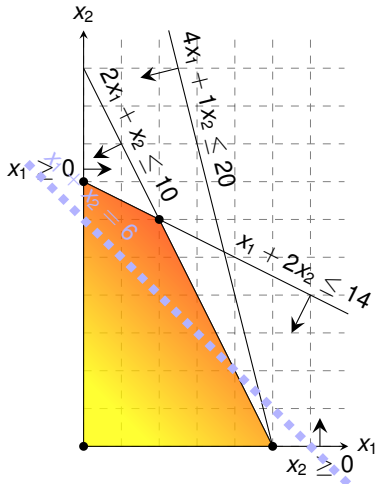
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far as possible.



Finding the Optimal Production Schedule

$$\begin{array}{llll} \text{maximise} & x_1 & + & x_2 \\ \text{subject to} & & & \\ & 4x_1 & + & x_2 \leq 20 \\ & 2x_1 & + & x_2 \leq 10 \\ & x_1 & + & 2x_2 \leq 14 \\ & x_1, x_2 & & \geq 0 \end{array}$$

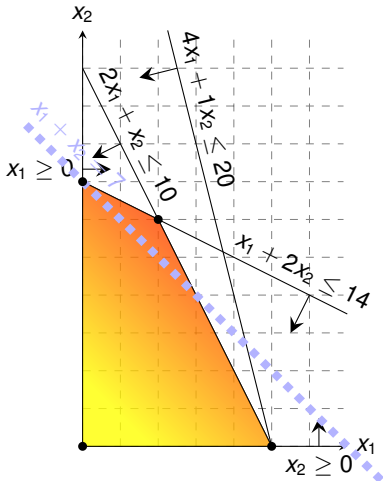
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



Finding the Optimal Production Schedule

$$\begin{array}{llll} \text{maximise} & x_1 & + & x_2 \\ \text{subject to} & & & \\ & 4x_1 & + & x_2 \leq 20 \\ & 2x_1 & + & x_2 \leq 10 \\ & x_1 & + & 2x_2 \leq 14 \\ & x_1, x_2 & & \geq 0 \end{array}$$

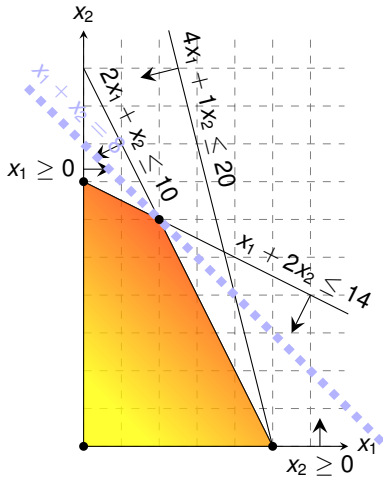
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



Finding the Optimal Production Schedule

$$\begin{array}{llll} \text{maximise} & x_1 & + & x_2 \\ \text{subject to} & & & \\ & 4x_1 & + & x_2 \leq 20 \\ & 2x_1 & + & x_2 \leq 10 \\ & x_1 & + & 2x_2 \leq 14 \\ & x_1, x_2 & & \geq 0 \end{array}$$

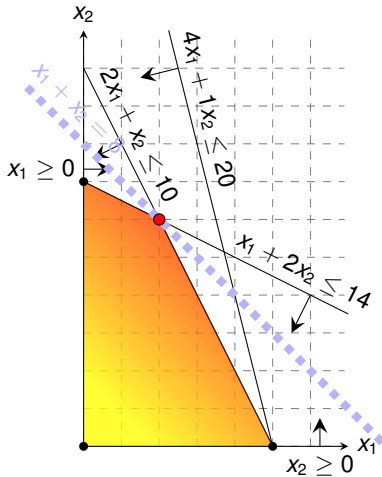
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



Finding the Optimal Production Schedule

$$\begin{array}{llll} \text{maximise} & x_1 & + & x_2 \\ \text{subject to} & & & \\ & 4x_1 & + & x_2 \leq 20 \\ & 2x_1 & + & x_2 \leq 10 \\ & x_1 & + & 2x_2 \leq 14 \\ & x_1, x_2 & & \geq 0 \end{array}$$

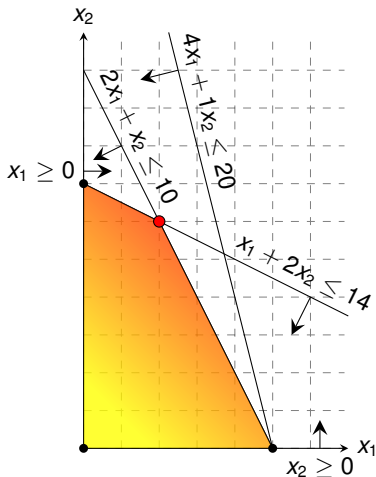
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



Finding the Optimal Production Schedule

$$\begin{array}{llll} \text{maximise} & x_1 & + & x_2 \\ \text{subject to} & & & \\ & 4x_1 & + & x_2 \leq 20 \\ & 2x_1 & + & x_2 \leq 10 \\ & x_1 & + & 2x_2 \leq 14 \\ & x_1, x_2 & & \geq 0 \end{array}$$

Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.

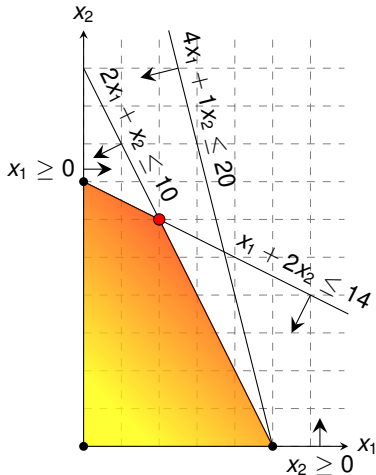


Question: Which aspect did we ignore in the formulation of the linear program?

Finding the Optimal Production Schedule

$$\begin{array}{llll} \text{maximise} & x_1 & + & x_2 \\ \text{subject to} & & & \\ & 4x_1 & + & x_2 \leq 20 \\ & 2x_1 & + & x_2 \leq 10 \\ & x_1 & + & 2x_2 \leq 14 \\ & x_1, x_2 & & \geq 0 \end{array}$$

Graphical Procedure: Move the line $x_1 + x_2 = z$ as far as possible.



While the same approach also works for higher-dimensions, we need to take a more systematic and algebraic procedure.

Outline

Introduction

A Simple Example of a Linear Program

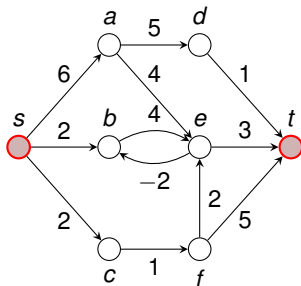
Formulating Problems as Linear Programs

Standard and Slack Forms

Shortest Paths

Single-Pair Shortest Path Problem

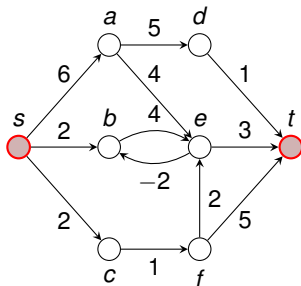
- Given: directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$



Shortest Paths

Single-Pair Shortest Path Problem

- **Given:** directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$
- **Goal:** Find a path of **minimum weight** from s to t in G

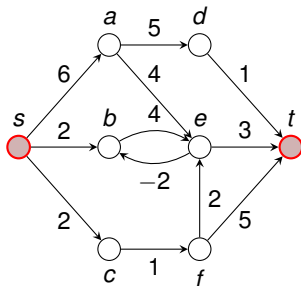


Shortest Paths

Single-Pair Shortest Path Problem

- **Given:** directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$
- **Goal:** Find a path of **minimum weight** from s to t in G

$p = (v_0 = s, v_1, \dots, v_k = t)$ such that $w(p) = \sum_{i=1}^k w(v_{i-1}, v_i)$ is **minimised**.

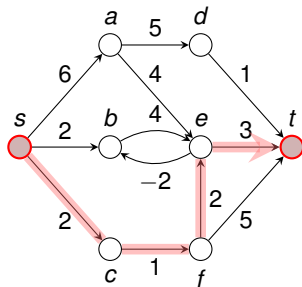


Shortest Paths

Single-Pair Shortest Path Problem

- **Given:** directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$
- **Goal:** Find a path of **minimum weight** from s to t in G

$p = (v_0 = s, v_1, \dots, v_k = t)$ such that $w(p) = \sum_{i=1}^k w(v_{i-1}, v_i)$ is **minimised**.

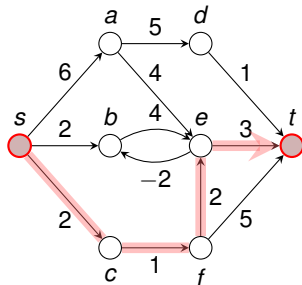


Shortest Paths

Single-Pair Shortest Path Problem

- **Given:** directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$
- **Goal:** Find a path of **minimum weight** from s to t in G

$p = (v_0 = s, v_1, \dots, v_k = t)$ such that $w(p) = \sum_{i=1}^k w(v_{k-1}, v_k)$ is **minimised**.



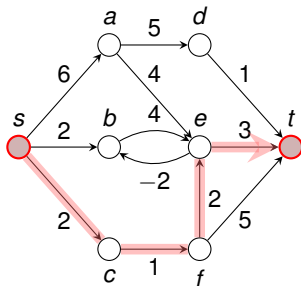
Exercise: Translate the SPSP problem into a linear program!

Shortest Paths

Single-Pair Shortest Path Problem

- **Given:** directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$
- **Goal:** Find a path of **minimum weight** from s to t in G

$p = (v_0 = s, v_1, \dots, v_k = t)$ such that $w(p) = \sum_{i=1}^k w(v_{k-1}, v_k)$ is **minimised**.



Shortest Paths as LP

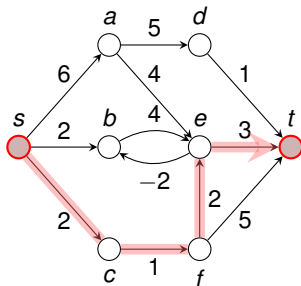
subject to

Shortest Paths

Single-Pair Shortest Path Problem

- **Given:** directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$
- **Goal:** Find a path of **minimum weight** from s to t in G

$p = (v_0 = s, v_1, \dots, v_k = t)$ such that $w(p) = \sum_{i=1}^k w(v_{k-1}, v_k)$ is **minimised**.



Shortest Paths as LP

subject to

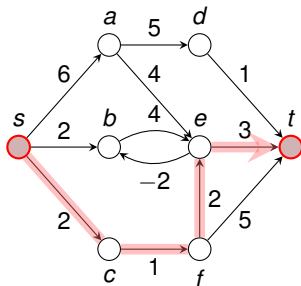
$$\begin{aligned} d_v &\leq d_u + w(u, v) && \text{for each edge } (u, v) \in E, \\ d_s &= 0. \end{aligned}$$

Shortest Paths

Single-Pair Shortest Path Problem

- **Given:** directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$
- **Goal:** Find a path of **minimum weight** from s to t in G

$p = (v_0 = s, v_1, \dots, v_k = t)$ such that $w(p) = \sum_{i=1}^k w(v_{k-1}, v_k)$ is **minimised**.



Shortest Paths as LP

maximise d_t

subject to

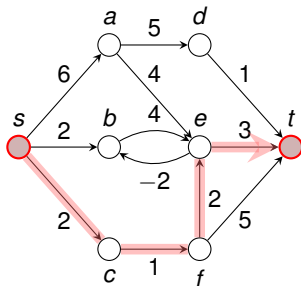
$$d_v \leq d_u + w(u, v) \quad \text{for each edge } (u, v) \in E,$$
$$d_s = 0.$$

Shortest Paths

Single-Pair Shortest Path Problem

- **Given:** directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$
- **Goal:** Find a path of **minimum weight** from s to t in G

$p = (v_0 = s, v_1, \dots, v_k = t)$ such that $w(p) = \sum_{i=1}^k w(v_{k-1}, v_k)$ is **minimised**.



Shortest Paths as LP

maximise
subject to

$$\begin{aligned} d_v &\leq d_u + w(u, v) && \text{for each edge } (u, v) \in E, \\ d_s &= 0. \end{aligned}$$

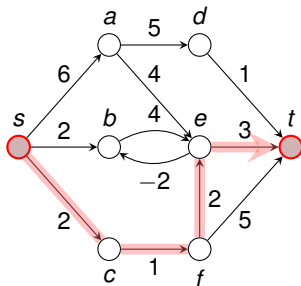
this is a **maximisation problem!**

Shortest Paths

Single-Pair Shortest Path Problem

- **Given:** directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$
- **Goal:** Find a path of **minimum weight** from s to t in G

$p = (v_0 = s, v_1, \dots, v_k = t)$ such that $w(p) = \sum_{i=1}^k w(v_{k-1}, v_k)$ is **minimised**.



Shortest Paths as LP

maximise d_t
subject to

$$d_v \leq d_u + w(u, v) \quad \text{for each edge } (u, v) \in E,$$
$$d_s = 0.$$

this is a **maximisation problem!**

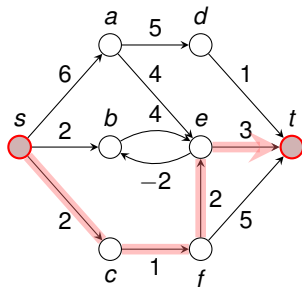
Recall: When BELLMAN-FORD terminates, all these inequalities are satisfied.

Shortest Paths

Single-Pair Shortest Path Problem

- **Given:** directed graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$, pair of vertices $s, t \in V$
- **Goal:** Find a path of **minimum weight** from s to t in G

$p = (v_0 = s, v_1, \dots, v_k = t)$ such that $w(p) = \sum_{i=1}^k w(v_{k-1}, v_k)$ is **minimised**.



Shortest Paths as LP

maximise d_t
subject to

$$d_v \leq d_u + w(u, v) \quad \text{for each edge } (u, v) \in E,$$
$$d_s = 0.$$

this is a **maximisation problem!**

Recall: When BELLMAN-FORD terminates, all these inequalities are satisfied.

Solution \bar{d} satisfies $\bar{d}_v = \min_{u: (u,v) \in E} \{\bar{d}_u + w(u, v)\}$

Maximum Flow

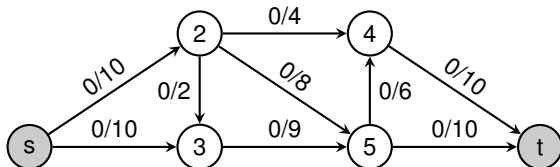
Maximum Flow Problem

- **Given:** directed graph $G = (V, E)$ with edge capacities $c : E \rightarrow \mathbb{R}^+$ (recall $c(u, v) = 0$ if $(u, v) \notin E$), pair of vertices $s, t \in V$

Maximum Flow

Maximum Flow Problem

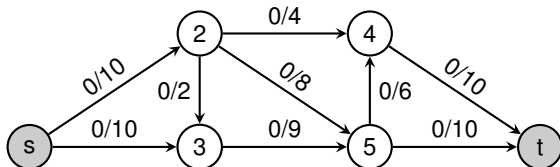
- Given: directed graph $G = (V, E)$ with edge capacities $c : E \rightarrow \mathbb{R}^+$ (recall $c(u, v) = 0$ if $(u, v) \notin E$), pair of vertices $s, t \in V$



Maximum Flow

Maximum Flow Problem

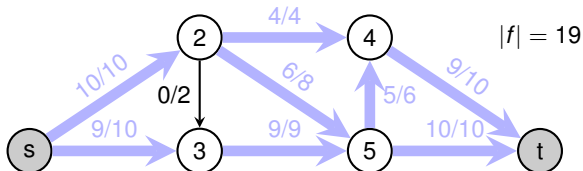
- **Given:** directed graph $G = (V, E)$ with edge capacities $c : E \rightarrow \mathbb{R}^+$ (recall $c(u, v) = 0$ if $(u, v) \notin E$), pair of vertices $s, t \in V$
- **Goal:** Find a **maximum flow** $f : V \times V \rightarrow \mathbb{R}$ from s to t which satisfies the capacity constraints and flow conservation



Maximum Flow

Maximum Flow Problem

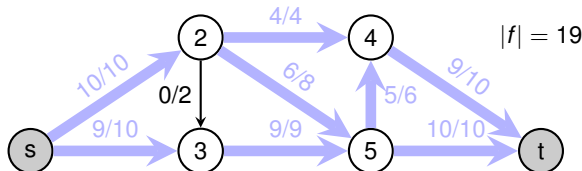
- **Given:** directed graph $G = (V, E)$ with edge capacities $c : E \rightarrow \mathbb{R}^+$ (recall $c(u, v) = 0$ if $(u, v) \notin E$), pair of vertices $s, t \in V$
- **Goal:** Find a **maximum flow** $f : V \times V \rightarrow \mathbb{R}$ from s to t which satisfies the capacity constraints and flow conservation



Maximum Flow

Maximum Flow Problem

- Given: directed graph $G = (V, E)$ with edge capacities $c : E \rightarrow \mathbb{R}^+$ (recall $c(u, v) = 0$ if $(u, v) \notin E$), pair of vertices $s, t \in V$
- Goal: Find a maximum flow $f : V \times V \rightarrow \mathbb{R}$ from s to t which satisfies the capacity constraints and flow conservation



Maximum Flow as LP

maximise
subject to

$$\sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs}$$

$$\begin{aligned} f_{uv} &\leq c(u, v) && \text{for each } u, v \in V, \\ \sum_{v \in V} f_{vu} &= \sum_{v \in V} f_{uv} && \text{for each } u \in V \setminus \{s, t\}, \\ f_{uv} &\geq 0 && \text{for each } u, v \in V. \end{aligned}$$

Minimum-Cost Flow

Extension of the Maximum Flow Problem

Minimum-Cost-Flow Problem



Minimum-Cost Flow

Extension of the Maximum Flow Problem

Minimum-Cost-Flow Problem

- **Given:** directed graph $G = (V, E)$ with capacities $c : E \rightarrow \mathbb{R}^+$, pair of vertices $s, t \in V$, **cost function** $a : E \rightarrow \mathbb{R}^+$, **flow demand of d units**

Minimum-Cost Flow

Extension of the Maximum Flow Problem

Minimum-Cost-Flow Problem

- **Given:** directed graph $G = (V, E)$ with capacities $c : E \rightarrow \mathbb{R}^+$, pair of vertices $s, t \in V$, **cost function** $a : E \rightarrow \mathbb{R}^+$, **flow demand of d units**
- **Goal:** Find a **flow** $f : V \times V \rightarrow \mathbb{R}$ from s to t with $|f| = d$ while **minimising the total cost** $\sum_{(u,v) \in E} a(u,v) f_{uv}$ incurred by the flow.

Minimum-Cost Flow

Extension of the Maximum Flow Problem

Minimum-Cost-Flow Problem

- **Given:** directed graph $G = (V, E)$ with capacities $c : E \rightarrow \mathbb{R}^+$, pair of vertices $s, t \in V$, **cost function** $a : E \rightarrow \mathbb{R}^+$, **flow demand** of d units
- **Goal:** Find a **flow** $f : V \times V \rightarrow \mathbb{R}$ from s to t with $|f| = d$ while **minimising the total cost** $\sum_{(u,v) \in E} a(u,v)f_{uv}$ incurred by the flow.

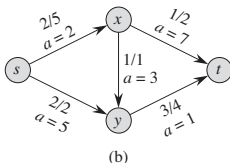
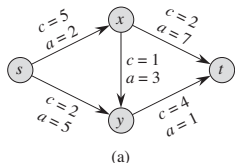


Figure 29.3 (a) An example of a minimum-cost-flow problem. We denote the capacities by c and the costs by a . Vertex s is the source and vertex t is the sink, and we wish to send 4 units of flow from s to t . (b) A solution to the minimum-cost flow problem in which 4 units of flow are sent from s to t . For each edge, the flow and capacity are written as flow/capacity.

Minimum-Cost Flow

Extension of the Maximum Flow Problem

Minimum-Cost-Flow Problem

- **Given:** directed graph $G = (V, E)$ with capacities $c : E \rightarrow \mathbb{R}^+$, pair of vertices $s, t \in V$, **cost function** $a : E \rightarrow \mathbb{R}^+$, **flow demand** of d units
- **Goal:** Find a **flow** $f : V \times V \rightarrow \mathbb{R}$ from s to t with $|f| = d$ while **minimising the total cost** $\sum_{(u,v) \in E} a(u,v)f_{uv}$ incurred by the flow.

Optimal Solution with total cost:

$$\sum_{(u,v) \in E} a(u,v)f_{uv} = (2 \cdot 2) + (5 \cdot 2) + (3 \cdot 1) + (7 \cdot 1) + (1 \cdot 3) = 27$$

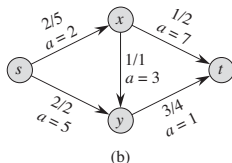
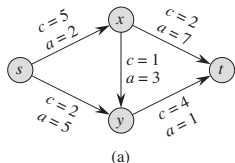


Figure 29.3 (a) An example of a minimum-cost-flow problem. We denote the capacities by c and the costs by a . Vertex s is the source and vertex t is the sink, and we wish to send 4 units of flow from s to t . (b) A solution to the minimum-cost flow problem in which 4 units of flow are sent from s to t . For each edge, the flow and capacity are written as flow/capacity.

Minimum Cost Flow as a LP

Minimum Cost Flow as LP

minimise $\sum_{(u,v) \in E} a(u,v) f_{uv}$

subject to

$$\begin{aligned} f_{uv} &\leq c(u,v) && \text{for } u, v \in V, \\ \sum_{v \in V} f_{vu} - \sum_{v \in V} f_{uv} &= 0 && \text{for } u \in V \setminus \{s, t\}, \\ \sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs} &= d, \\ f_{uv} &\geq 0 && \text{for } u, v \in V. \end{aligned}$$

Minimum Cost Flow as a LP

Minimum Cost Flow as LP

minimise $\sum_{(u,v) \in E} a(u,v) f_{uv}$

subject to

$$\begin{aligned} f_{uv} &\leq c(u,v) && \text{for } u, v \in V, \\ \sum_{v \in V} f_{vu} - \sum_{v \in V} f_{uv} &= 0 && \text{for } u \in V \setminus \{s, t\}, \\ \sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs} &= d, \\ f_{uv} &\geq 0 && \text{for } u, v \in V. \end{aligned}$$

Real power of Linear Programming comes from the ability to solve **new problems!**

Outline

Introduction

A Simple Example of a Linear Program

Formulating Problems as Linear Programs

Standard and Slack Forms

Standard and Slack Forms

Standard Form

maximise $\sum_{j=1}^n c_j x_j$

subject to

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \dots, m$$

$$x_j \geq 0 \quad \text{for } j = 1, 2, \dots, n$$

Standard and Slack Forms

Standard Form

maximise $\sum_{j=1}^n c_j x_j$ 

subject to

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \dots, m$$

$$x_j \geq 0 \quad \text{for } j = 1, 2, \dots, n$$

Standard and Slack Forms

Standard Form

maximise $\sum_{j=1}^n c_j x_j$ Objective Function

subject to

$n + m$ constraints

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

Standard and Slack Forms

Standard Form

maximise $\sum_{j=1}^n c_j x_j$ Objective Function

subject to

$n + m$ constraints $\left\{ \begin{array}{l} \sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \dots, m \\ x_j \geq 0 \quad \text{for } j = 1, 2, \dots, n \end{array} \right.$

Non-Negativity Constraints

Standard and Slack Forms

Standard Form

maximise $\sum_{j=1}^n c_j x_j$ Objective Function

subject to

$n + m$ constraints

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

Non-Negativity Constraints

Standard Form (Matrix-Vector-Notation)

maximise $c^T x$ Inner product of two vectors

subject to

$$\begin{array}{l} Ax \leq b \\ x \geq 0 \end{array} \quad \text{Matrix-vector product}$$

Converting Linear Programs into Standard Form

Reasons for a LP not being in standard form:

1. The objective might be a **minimisation** rather than **maximisation**.
2. There might be variables without **nonnegativity constraints**.
3. There might be **equality constraints**.
4. There might be **inequality constraints** (with \geq instead of \leq).

Converting Linear Programs into Standard Form

Reasons for a LP not being in standard form:

1. The objective might be a **minimisation** rather than **maximisation**.
2. There might be variables without **nonnegativity constraints**.
3. There might be **equality constraints**.
4. There might be **inequality constraints** (with \geq instead of \leq).

Goal: Convert linear program into an **equivalent** program which is in standard form

Converting Linear Programs into Standard Form

Reasons for a LP not being in standard form:

1. The objective might be a **minimisation** rather than **maximisation**.
2. There might be variables without **nonnegativity constraints**.
3. There might be **equality constraints**.
4. There might be **inequality constraints** (with \geq instead of \leq).

Goal: Convert linear program into an **equivalent** program which is in standard form

Equivalence: a correspondence (not necessarily a bijection) between solutions.

Converting into Standard Form (1/5)

Reasons for a LP not being in standard form:

1. The objective might be a **minimisation** rather than **maximisation**.

Converting into Standard Form (1/5)

Reasons for a LP not being in standard form:

1. The objective might be a **minimisation** rather than **maximisation**.

$$\text{minimise } -2x_1 + 3x_2$$

subject to

$$x_1 + x_2 = 7$$

$$x_1 - 2x_2 \leq 4$$

$$x_1 \geq 0$$

Converting into Standard Form (1/5)

Reasons for a LP not being in standard form:

1. The objective might be a **minimisation** rather than **maximisation**.

$$\text{minimise } -2x_1 + 3x_2$$

subject to

$$x_1 + x_2 = 7$$

$$x_1 - 2x_2 \leq 4$$

$$x_1 \geq 0$$



Negate objective function

Converting into Standard Form (1/5)

Reasons for a LP not being in standard form:

1. The objective might be a **minimisation** rather than **maximisation**.

$$\begin{array}{l} \text{minimise} \quad -2x_1 + 3x_2 \\ \text{subject to} \end{array}$$

$$\begin{array}{rclcl} x_1 & + & x_2 & = & 7 \\ x_1 & - & 2x_2 & \leq & 4 \\ x_1 & & & \geq & 0 \end{array}$$

Negate objective function

$$\begin{array}{l} \text{maximise} \quad 2x_1 - 3x_2 \\ \text{subject to} \end{array}$$

$$\begin{array}{rclcl} x_1 & + & x_2 & = & 7 \\ x_1 & - & 2x_2 & \leq & 4 \\ x_1 & & & \geq & 0 \end{array}$$

Converting into Standard Form (2/5)

Reasons for a LP not being in standard form:

2. There might be variables without nonnegativity constraints.

Converting into Standard Form (2/5)

Reasons for a LP not being in standard form:

2. There might be variables without nonnegativity constraints.

$$\begin{array}{ll} \text{maximise} & 2x_1 - 3x_2 \\ \text{subject to} & \\ & x_1 + x_2 = 7 \\ & x_1 - 2x_2 \leq 4 \\ & x_1 \geq 0 \end{array}$$

Converting into Standard Form (2/5)

Reasons for a LP not being in standard form:

2. There might be variables without nonnegativity constraints.

maximise
subject to

$$2x_1 - 3x_2$$

$$x_1 + x_2 = 7$$

$$x_1 - 2x_2 \leq 4$$

$$x_1 \geq 0$$



Replace x_2 by two non-negative variables x_2' and x_2''

Converting into Standard Form (2/5)

Reasons for a LP not being in standard form:

2. There might be variables without nonnegativity constraints.

maximise
subject to

$$2x_1 - 3x_2$$

$$x_1 + x_2 = 7$$

$$x_1 - 2x_2 \leq 4$$

$$x_1 \geq 0$$



Replace x_2 by two non-negative variables x'_2 and x''_2

maximise
subject to

$$2x_1 - 3x'_2 + 3x''_2$$

$$x_1 + x'_2 - x''_2 = 7$$

$$x_1 - 2x'_2 + 2x''_2 \leq 4$$

$$x_1, x'_2, x''_2 \geq 0$$

Converting into Standard Form (3/5)

Reasons for a LP not being in standard form:

3. There might be equality constraints.

Converting into Standard Form (3/5)

Reasons for a LP not being in standard form:

3. There might be equality constraints.

maximise
subject to

$$2x_1 - 3x_2' + 3x_2''$$

$$x_1 + x_2' - x_2'' = 7$$

$$x_1 - 2x_2' + 2x_2'' \leq 4$$

$$x_1, x_2', x_2'' \geq 0$$

Converting into Standard Form (3/5)

Reasons for a LP not being in standard form:

3. There might be equality constraints.

maximise
subject to

$$2x_1 - 3x_2' + 3x_2''$$

$$x_1 + x_2' - x_2'' = 7$$

$$x_1 - 2x_2' + 2x_2'' \leq 4$$

$$x_1, x_2', x_2'' \geq 0$$

↓ Replace each equality
by two inequalities.

Converting into Standard Form (3/5)

Reasons for a LP not being in standard form:

3. There might be equality constraints.

maximise
subject to

$$2x_1 - 3x_2' + 3x_2''$$

$$\begin{array}{rclcl} x_1 + x_2' - x_2'' & = & 7 \\ x_1 - 2x_2' + 2x_2'' & \leq & 4 \\ x_1, x_2', x_2'' & \geq & 0 \end{array}$$

Replace each equality
by two inequalities.

maximise
subject to

$$2x_1 - 3x_2' + 3x_2''$$

$$\begin{array}{rclcl} x_1 + x_2' - x_2'' & \leq & 7 \\ x_1 + x_2' - x_2'' & \geq & 7 \\ x_1 - 2x_2' + 2x_2'' & \leq & 4 \\ x_1, x_2', x_2'' & \geq & 0 \end{array}$$

Converting into Standard Form (4/5)

Reasons for a LP not being in standard form:

4. There might be inequality constraints (with \geq instead of \leq).

Converting into Standard Form (4/5)

Reasons for a LP not being in standard form:

4. There might be inequality constraints (with \geq instead of \leq).

maximise
subject to

$$2x_1 - 3x_2' + 3x_2''$$

$$x_1 + x_2' - x_2'' \leq 7$$

$$x_1 + x_2' - x_2'' \geq 7$$

$$x_1 - 2x_2' + 2x_2'' \leq 4$$

$$x_1, x_2', x_2'' \geq 0$$

Converting into Standard Form (4/5)

Reasons for a LP not being in standard form:

4. There might be inequality constraints (with \geq instead of \leq).

maximise
subject to

$$2x_1 - 3x_2' + 3x_2''$$

$$x_1 + x_2' - x_2'' \leq 7$$

$$x_1 + x_2' - x_2'' \geq 7$$

$$x_1 - 2x_2' + 2x_2'' \leq 4$$

$$x_1, x_2', x_2'' \geq 0$$

↓ Negate respective inequalities.

Converting into Standard Form (4/5)

Reasons for a LP not being in standard form:

4. There might be inequality constraints (with \geq instead of \leq).

maximise
subject to

$$\begin{array}{rcccccc} 2x_1 & - & 3x_2' & + & 3x_2'' & & \\ x_1 & + & x_2' & - & x_2'' & \leq & 7 \\ x_1 & + & x_2' & - & x_2'' & \geq & 7 \\ x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\ x_1, x_2', x_2'' & & & & & \geq & 0 \end{array}$$

Negate respective inequalities.

maximise
subject to

$$\begin{array}{rcccccc} 2x_1 & - & 3x_2' & + & 3x_2'' & & \\ x_1 & + & x_2' & - & x_2'' & \leq & 7 \\ -x_1 & - & x_2' & + & x_2'' & \leq & -7 \\ x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\ x_1, x_2', x_2'' & & & & & \geq & 0 \end{array}$$

Converting into Standard Form (5/5)

Rename variable names (for consistency).

maximise
subject to

$$2x_1 - 3x_2 + 3x_3$$

$$x_1 + x_2 - x_3 \leq 7$$

$$-x_1 - x_2 + x_3 \leq -7$$

$$x_1 - 2x_2 + 2x_3 \leq 4$$

$$x_1, x_2, x_3 \geq 0$$

Converting into Standard Form (5/5)

Rename variable names (for consistency).

$$\begin{array}{rcllcl} \text{maximise} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\ \text{subject to} & & & & & & & \\ & x_1 & + & x_2 & - & x_3 & \leq & 7 \\ & -x_1 & - & x_2 & + & x_3 & \leq & -7 \\ & x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\ & x_1, x_2, x_3 & & & & & \geq & 0 \end{array}$$

It is always possible to convert a linear program into standard form.

Converting Standard Form into Slack Form (1/3)

Goal: Convert *standard form* into *slack form*, where all constraints except for the non-negativity constraints are equalities.

Converting Standard Form into Slack Form (1/3)

Goal: Convert **standard form** into **slack form**, where all constraints except for the non-negativity constraints are equalities.

For the **simplex algorithm**, it is more convenient to work with equality constraints.

Converting Standard Form into Slack Form (1/3)

Goal: Convert **standard form** into **slack form**, where all constraints except for the non-negativity constraints are equalities.

For the **simplex algorithm**, it is more convenient to work with equality constraints.

Introducing Slack Variables



Converting Standard Form into Slack Form (1/3)

Goal: Convert **standard form** into **slack form**, where all constraints except for the non-negativity constraints are equalities.

For the **simplex algorithm**, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^n a_{ij}x_j \leq b_i$ be an inequality constraint

Converting Standard Form into Slack Form (1/3)

Goal: Convert **standard form** into **slack form**, where all constraints except for the non-negativity constraints are equalities.

For the **simplex algorithm**, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^n a_{ij}x_j \leq b_i$ be an inequality constraint
- Introduce a **slack variable** s by

Converting Standard Form into Slack Form (1/3)

Goal: Convert **standard form** into **slack form**, where all constraints except for the non-negativity constraints are equalities.

For the **simplex algorithm**, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^n a_{ij}x_j \leq b_i$ be an inequality constraint
- Introduce a **slack variable** s by

$$s = b_i - \sum_{j=1}^n a_{ij}x_j$$

Converting Standard Form into Slack Form (1/3)

Goal: Convert **standard form** into **slack form**, where all constraints except for the non-negativity constraints are equalities.

For the **simplex algorithm**, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^n a_{ij}x_j \leq b_i$ be an inequality constraint
- Introduce a **slack variable** s by

$$s = b_i - \sum_{j=1}^n a_{ij}x_j$$

$$s \geq 0.$$

Converting Standard Form into Slack Form (1/3)

Goal: Convert **standard form** into **slack form**, where all constraints except for the non-negativity constraints are equalities.

For the **simplex algorithm**, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^n a_{ij}x_j \leq b_i$ be an inequality constraint
- Introduce a **slack variable** s by

s measures the slack between the two sides of the inequality.

$$s = b_i - \sum_{j=1}^n a_{ij}x_j$$

$$s \geq 0.$$

Converting Standard Form into Slack Form (1/3)

Goal: Convert **standard form** into **slack form**, where all constraints except for the non-negativity constraints are equalities.

For the **simplex algorithm**, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^n a_{ij}x_j \leq b_i$ be an inequality constraint
- Introduce a **slack variable** s by

s measures the slack between the two sides of the inequality.

$$s = b_i - \sum_{j=1}^n a_{ij}x_j$$
$$s \geq 0.$$

- Denote slack variable of the i -th inequality by x_{n+i}

Converting Standard Form into Slack Form (2/3)

$$\begin{array}{ll} \text{maximise} & 2x_1 - 3x_2 + 3x_3 \\ \text{subject to} & \\ & x_1 + x_2 - x_3 \leq 7 \\ & -x_1 - x_2 + x_3 \leq -7 \\ & x_1 - 2x_2 + 2x_3 \leq 4 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Converting Standard Form into Slack Form (2/3)

maximise
subject to

$$\begin{array}{rcccccc} 2x_1 & - & 3x_2 & + & 3x_3 & & \\ x_1 & + & x_2 & - & x_3 & \leq & 7 \\ -x_1 & - & x_2 & + & x_3 & \leq & -7 \\ x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\ & & & & & \geq & 0 \end{array}$$

x_1, x_2, x_3



Introduce slack variables

Converting Standard Form into Slack Form (2/3)

maximise
subject to

$$\begin{array}{rcccccc} 2x_1 & - & 3x_2 & + & 3x_3 & & \\ x_1 & + & x_2 & - & x_3 & \leq & 7 \\ -x_1 & - & x_2 & + & x_3 & \leq & -7 \\ x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\ & & & & & \geq & 0 \end{array}$$

x_1, x_2, x_3

Introduce slack variables

subject to

$$x_4 = 7 - x_1 - x_2 + x_3$$

Converting Standard Form into Slack Form (2/3)

maximise
subject to

$$\begin{array}{rcccccc} 2x_1 & - & 3x_2 & + & 3x_3 & & \\ x_1 & + & x_2 & - & x_3 & \leq & 7 \\ -x_1 & - & x_2 & + & x_3 & \leq & -7 \\ x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\ & & & & & \geq & 0 \end{array}$$

x_1, x_2, x_3

Introduce slack variables

subject to

$$\begin{array}{rccccccccc} x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \end{array}$$

Converting Standard Form into Slack Form (2/3)

maximise
subject to

$$\begin{array}{rcccccc} 2x_1 & - & 3x_2 & + & 3x_3 & & \\ x_1 & + & x_2 & - & x_3 & \leq & 7 \\ -x_1 & - & x_2 & + & x_3 & \leq & -7 \\ x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\ & & & & & \geq & 0 \end{array}$$

x_1, x_2, x_3

Introduce slack variables

subject to

$$\begin{array}{rcccccc} x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\ x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \end{array}$$

Converting Standard Form into Slack Form (2/3)

maximise
subject to

$$\begin{array}{rcccccc} 2x_1 & - & 3x_2 & + & 3x_3 & & \\ x_1 & + & x_2 & - & x_3 & \leq & 7 \\ -x_1 & - & x_2 & + & x_3 & \leq & -7 \\ x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\ & & & & & \geq & 0 \end{array}$$

x_1, x_2, x_3

Introduce slack variables

subject to

$$\begin{array}{rccccccccc} x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\ x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \\ & & & & & & & & & \geq & 0 \end{array}$$

$x_1, x_2, x_3, x_4, x_5, x_6$

Converting Standard Form into Slack Form (2/3)

maximise
subject to

$$\begin{array}{rcccccc} 2x_1 & - & 3x_2 & + & 3x_3 & & \\ x_1 & + & x_2 & - & x_3 & \leq & 7 \\ -x_1 & - & x_2 & + & x_3 & \leq & -7 \\ x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\ & & & & & \geq & 0 \end{array}$$

x_1, x_2, x_3

Introduce slack variables

maximise
subject to

$$\begin{array}{rcccccccc} 2x_1 & - & 3x_2 & + & 3x_3 & & & \\ x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\ x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \\ & & & & & & & \geq & 0 \end{array}$$

$x_1, x_2, x_3, x_4, x_5, x_6$

Converting Standard Form into Slack Form (3/3)

$$\begin{array}{rcllclclcl} \text{maximise} & 2x_1 & - & 3x_2 & + & 3x_3 & & & \\ \text{subject to} & & & & & & & & \\ & x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ & x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\ & x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \\ & & & x_1, x_2, x_3, x_4, x_5, x_6 & & & \geq & 0 & & \end{array}$$

Converting Standard Form into Slack Form (3/3)

maximise
subject to

$$2x_1 - 3x_2 + 3x_3$$

$$x_4 = 7 - x_1 - x_2 + x_3$$

$$x_5 = -7 + x_1 + x_2 - x_3$$

$$x_6 = 4 - x_1 + 2x_2 - 2x_3$$

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



Use variable z to denote objective function and omit the nonnegativity constraints.

Converting Standard Form into Slack Form (3/3)

maximise
subject to

$$2x_1 - 3x_2 + 3x_3$$

$$x_4 = 7 - x_1 - x_2 + x_3$$

$$x_5 = -7 + x_1 + x_2 - x_3$$

$$x_6 = 4 - x_1 + 2x_2 - 2x_3$$

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

Use variable z to denote objective function
and omit the nonnegativity constraints.

$$\begin{array}{rcl} z & = & 2x_1 - 3x_2 + 3x_3 \\ x_4 & = & 7 - x_1 - x_2 + x_3 \\ x_5 & = & -7 + x_1 + x_2 - x_3 \\ x_6 & = & 4 - x_1 + 2x_2 - 2x_3 \end{array}$$

Converting Standard Form into Slack Form (3/3)

maximise
subject to

$$2x_1 - 3x_2 + 3x_3$$

$$x_4 = 7 - x_1 - x_2 + x_3$$

$$x_5 = -7 + x_1 + x_2 - x_3$$

$$x_6 = 4 - x_1 + 2x_2 - 2x_3$$

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

Use variable z to denote objective function
and omit the nonnegativity constraints.

$$\begin{array}{rcl} z & = & 2x_1 - 3x_2 + 3x_3 \\ x_4 & = & 7 - x_1 - x_2 + x_3 \\ x_5 & = & -7 + x_1 + x_2 - x_3 \\ x_6 & = & 4 - x_1 + 2x_2 - 2x_3 \end{array}$$

This is called **slack form**.

Basic and Non-Basic Variables

$$\begin{array}{rclclclcl} Z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\ x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\ x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \end{array}$$

Basic and Non-Basic Variables

$$\begin{array}{rcccccccc} Z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\ x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\ x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \end{array}$$

Basic Variables: $B = \{4, 5, 6\}$

Basic and Non-Basic Variables

$$\begin{array}{rcccccccc} Z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\ x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\ x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \end{array}$$

Basic Variables: $B = \{4, 5, 6\}$

Non-Basic Variables: $N = \{1, 2, 3\}$

Basic and Non-Basic Variables

$$\begin{array}{rccccccc} z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\ x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\ x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \end{array}$$

Basic Variables: $B = \{4, 5, 6\}$

Non-Basic Variables: $N = \{1, 2, 3\}$

Slack Form (Formal Definition)

Slack form is given by a tuple (N, B, A, b, c, v) so that

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

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

and all variables are non-negative.

Basic and Non-Basic Variables

$$\begin{array}{rccccccc} z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\ x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\ x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \end{array}$$

Basic Variables: $B = \{4, 5, 6\}$

Non-Basic Variables: $N = \{1, 2, 3\}$

Slack Form (Formal Definition)

Slack form is given by a tuple (N, B, A, b, c, v) so that

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

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

and all variables are non-negative.

Variables/Coefficients on the right hand side are indexed by B and N .

Slack Form (Example)

$$\begin{array}{rclclclcl} z & = & 28 & - & \frac{x_3}{6} & - & \frac{x_5}{6} & - & \frac{2x_6}{3} \\ x_1 & = & 8 & + & \frac{x_3}{6} & + & \frac{x_5}{6} & - & \frac{x_6}{3} \\ x_2 & = & 4 & - & \frac{8x_3}{3} & - & \frac{2x_5}{3} & + & \frac{x_6}{3} \\ x_4 & = & 18 & - & \frac{x_3}{2} & + & \frac{x_5}{2} & & \end{array}$$

Slack Form (Example)

$$\begin{aligned}z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\x_2 &= 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2}\end{aligned}$$

Slack Form Notation



Slack Form (Example)

$$\begin{array}{rclclclcl} z & = & 28 & - & \frac{x_3}{6} & - & \frac{x_5}{6} & - & \frac{2x_6}{3} \\ x_1 & = & 8 & + & \frac{x_3}{6} & + & \frac{x_5}{6} & - & \frac{x_6}{3} \\ x_2 & = & 4 & - & \frac{8x_3}{3} & - & \frac{2x_5}{3} & + & \frac{x_6}{3} \\ x_4 & = & 18 & - & \frac{x_3}{2} & + & \frac{x_5}{2} & & \end{array}$$

Slack Form Notation

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

Slack Form (Example)

$$\begin{aligned}z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\x_2 &= 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2}\end{aligned}$$

Slack Form Notation

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

-

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

Slack Form (Example)

$$\begin{aligned}z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\x_2 &= 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2}\end{aligned}$$

Slack Form Notation

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

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

- $$b = \begin{pmatrix} b_1 \\ b_2 \\ b_4 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 18 \end{pmatrix},$$

Slack Form (Example)

$$\begin{aligned}z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\x_2 &= 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2}\end{aligned}$$

Slack Form Notation

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

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

- $$b = \begin{pmatrix} b_1 \\ b_2 \\ b_4 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 18 \end{pmatrix}, \quad c = \begin{pmatrix} c_3 \\ c_5 \\ c_6 \end{pmatrix} = \begin{pmatrix} -1/6 \\ -1/6 \\ -2/3 \end{pmatrix}$$

Slack Form (Example)

$$\begin{aligned}z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\x_2 &= 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2}\end{aligned}$$

Slack Form Notation

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

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

- $$b = \begin{pmatrix} b_1 \\ b_2 \\ b_4 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 18 \end{pmatrix}, c = \begin{pmatrix} c_3 \\ c_5 \\ c_6 \end{pmatrix} = \begin{pmatrix} -1/6 \\ -1/6 \\ -2/3 \end{pmatrix}$$

- $v = 28$