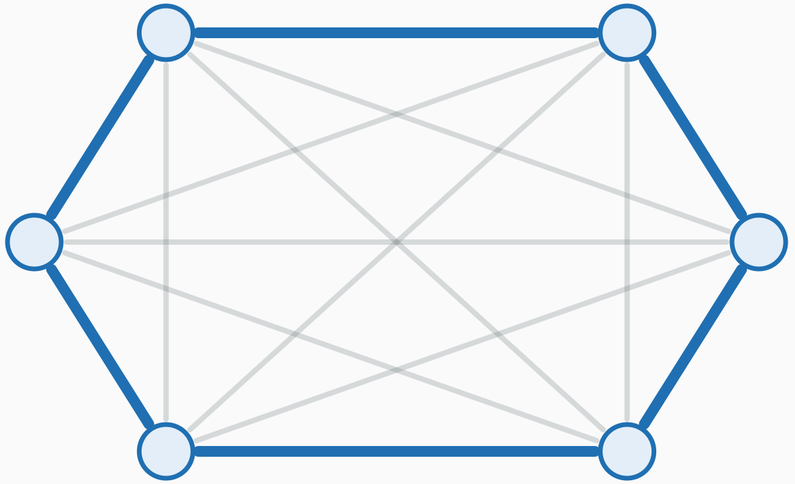
ALGORITHMS AND DATA STRUCTURES · LECTURE 16

Modelling with linear programs

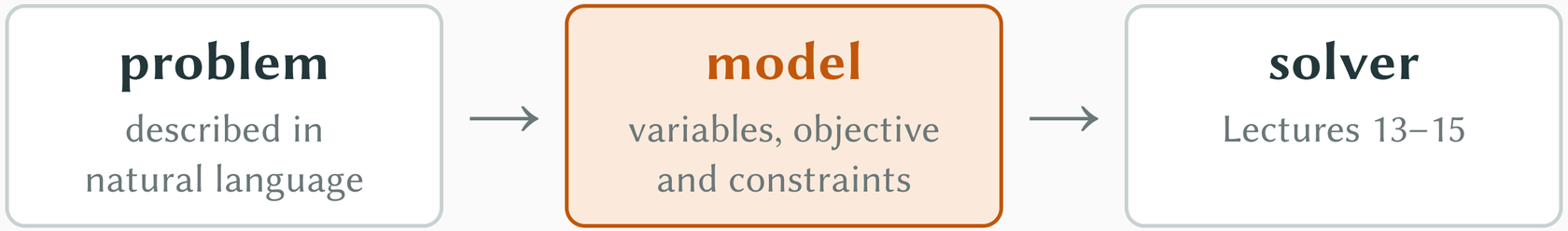


Heng Guo

School of Informatics, University of Edinburgh

Thursday 12 November 2026 · 16:10–17:00

From solving to modelling



Lectures 13 to 15: how to **solve** a linear program.

Today: how to **write one down**.

You will likely never need to solve an LP by hand after this class.
In practice, use a solver (CPLEX, Gurobi, ...).

? *Which problems can we model, and how?*

A recipe for modelling

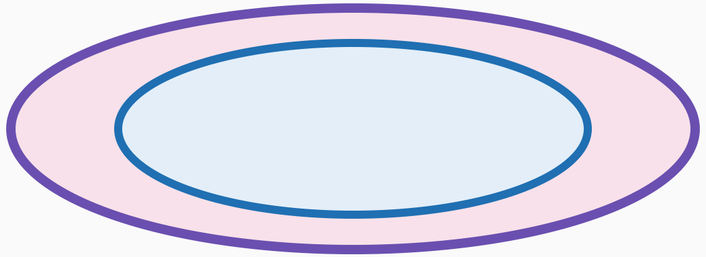
1. **Variables.** What do we decide?
   Everything else is data.
2. **Objective.** What do we maximise or minimise?
   A linear function of the variables.
3. **Constraints.** What must hold?
   Linear inequalities and equations.

blue: valid solutions
purple: what the model allows
magenta: the mismatch

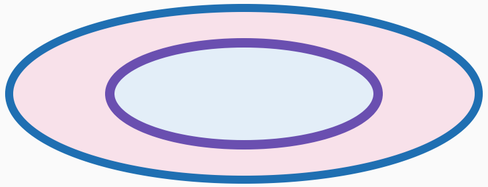
Check: the model must allow exactly the valid solutions.



correct:
feasible points = valid solutions



too weak:
a constraint is missing



too strict:
valid solutions are forbidden

Managing a production facility

A linear program

The problem

A factory makes **dark** and **milk chocolate** from **cocoa**, **sugar** and **milk powder**.

Goal: maximise the revenue:
the price minus the cost of the materials.

? *How much of each chocolate should we make?*

	dark $j=1$	milk $j=2$	stock b_i	unit cost c_i
cocoa $i=1$	2	1	8	£1
sugar $i=2$	1	2	6	£1
milk powder $i=3$	0	2	5	£2
price p_j	£6	£9		

a_{ij} = units of material i for one unit of product j.
Here $n = 2$ products and $m = 3$ materials.

Variables and objective

STEP 1 **Variables:** x_j = units of product j made.

x_1 dark, x_2 milk chocolate.

Any amount, also fractions.

The numbers a_{ij}, b_i, c_i, p_j are data.

STEP 2 **Objective:** revenue of one unit of product j

$$r_j = p_j - \sum_{i=1}^{m} a_{ij}\, c_i$$

its price minus the cost of its materials

dark: $r_1 = 6 - (2 \cdot 1 + 1 \cdot 1 + 0 \cdot 2) = 3$

milk: $r_2 = 9 - (1 \cdot 1 + 2 \cdot 1 + 2 \cdot 2) = 2$

Objective: maximise $3x_1 + 2x_2 = \sum_{j=1}^{n} r_j x_j$

	dark $j=1$	milk $j=2$	stock b_i	unit cost c_i
cocoa $i=1$	2	1	8	£1
sugar $i=2$	1	2	6	£1
milk powder $i=3$	0	2	5	£2
price p_j	£6	£9		

The constraints

STEP 3 **Constraints:** what limits the plan?

$x_j \geqslant 0$ for every product j

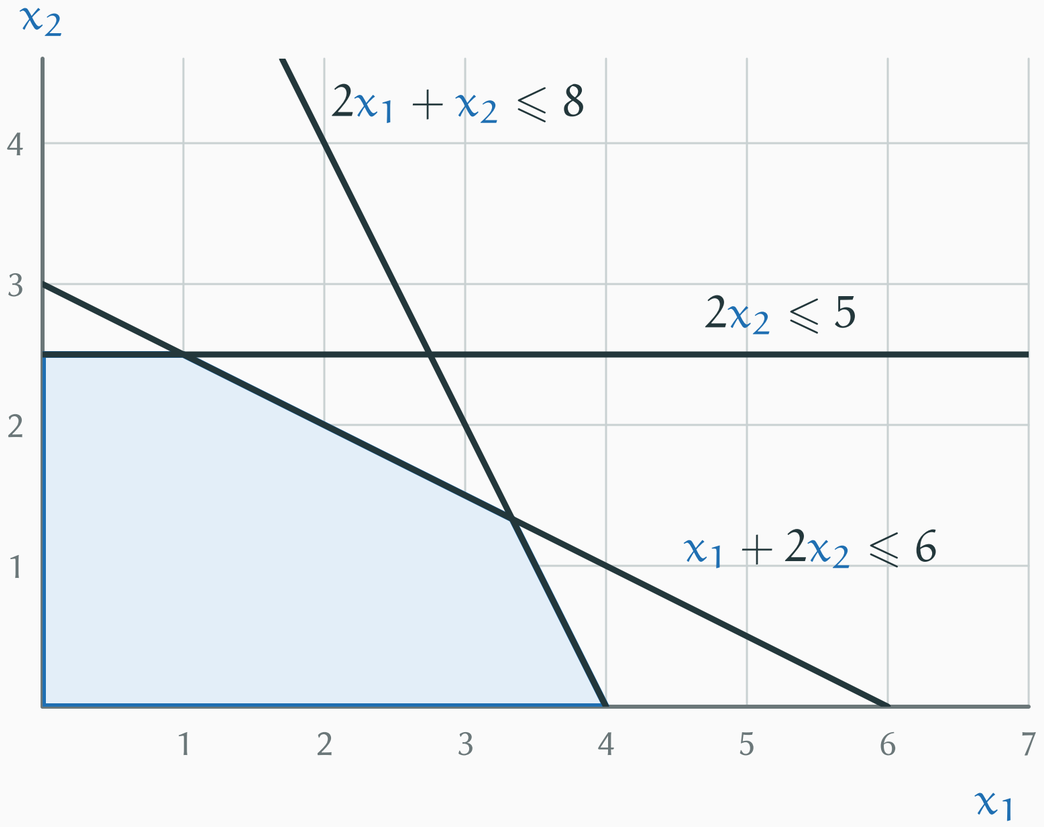
we cannot make a negative amount

$\sum_{j=1}^{n} a_{ij}\, x_j \leqslant b_i$ for every material i

use at most what is in stock

cocoa	$2x_1 + x_2 \leqslant 8$
sugar	$x_1 + 2x_2 \leqslant 6$
milk powder	$2x_2 \leqslant 5$

Standard form: maximise $r^\top x$
subject to $Ax \leqslant b$, $x \geqslant 0$.



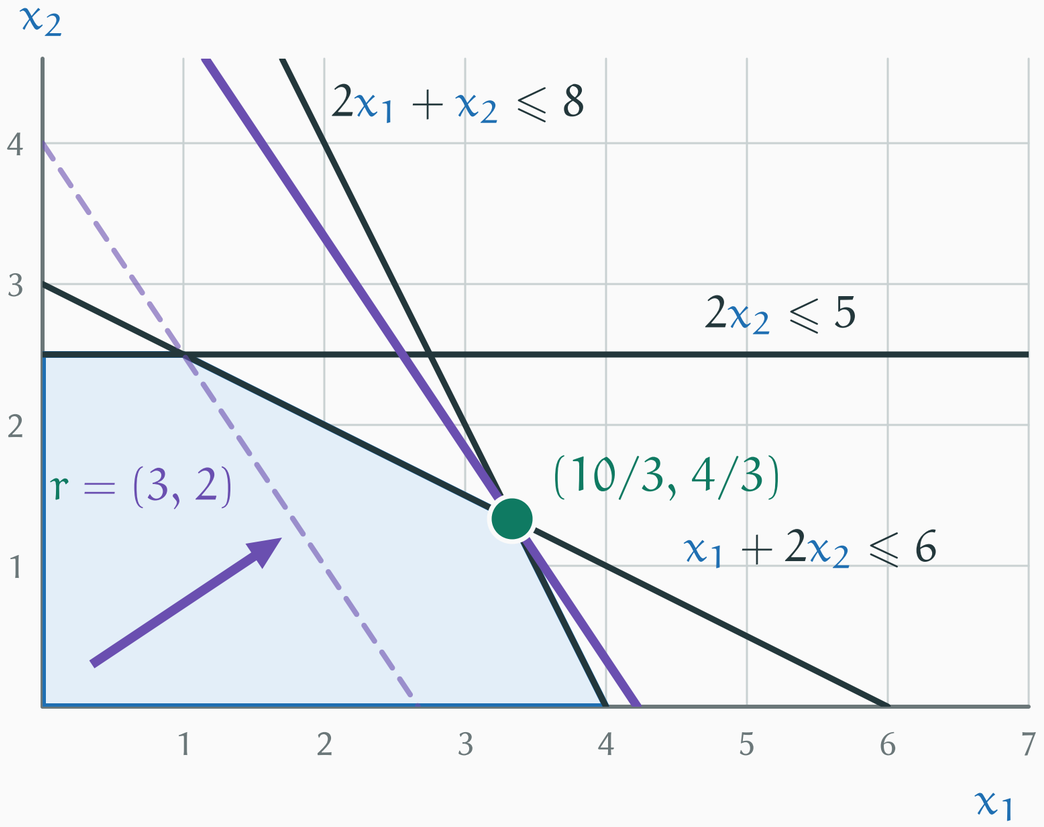
Solved

$$
\begin{aligned}
\text{maximise} \quad & 3x_1 + 2x_2 \\
\text{subject to} \quad & 2x_1 + x_2 \leqslant 8 \\
& x_1 + 2x_2 \leqslant 6 \\
& 2x_2 \leqslant 5 \\
& x_1, x_2 \geqslant 0
\end{aligned}
$$

Revenue grows in the direction $r = (3, 2)$.

Optimum $x^* = (10/3, 4/3)$, revenue $38/3$.

Cocoa 8 of 8, sugar 6 of 6: used up.
Milk powder 8/3 of 5.



Integer programs

Integer programs and 0–1 variables

ILP: an LP whose variables must be **integers**.

Same numbers, but chairs and sofas: they come whole, so x_1, x_2 are integers.

Feasible: the **12 integer points** of the polygon.

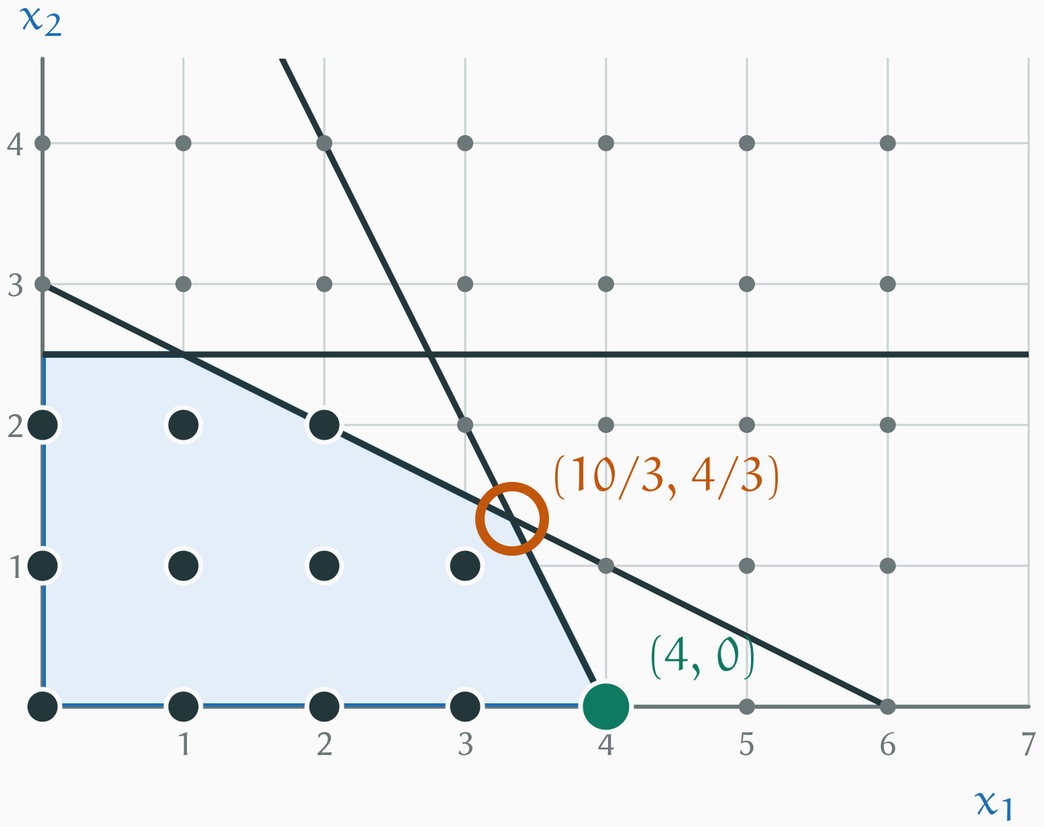
LP optimum $(10/3, 4/3)$: $38/3$

ILP optimum $(4, 0)$: 12

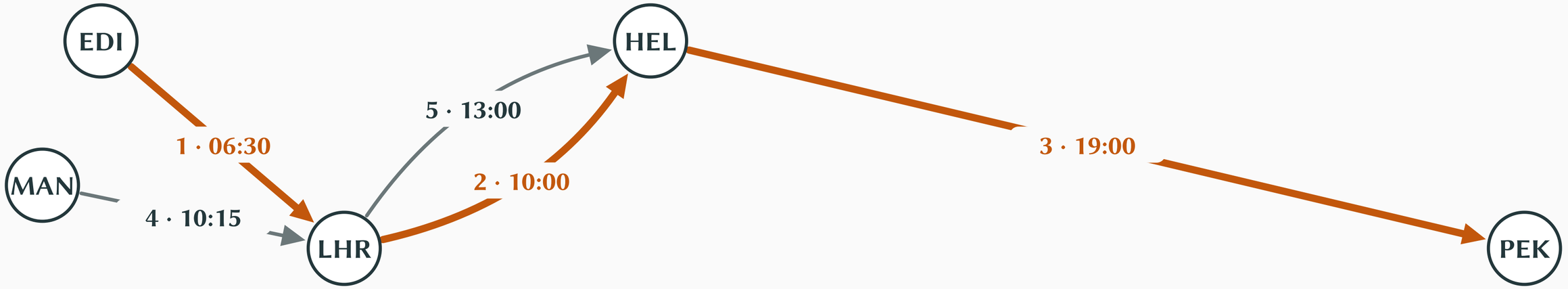
Rounding the LP optimum gives only $(3, 1)$: 11.

0–1 variables.

$x_j \in \{0, 1\}$ answers a yes/no question.



Flight scheduling: the problem



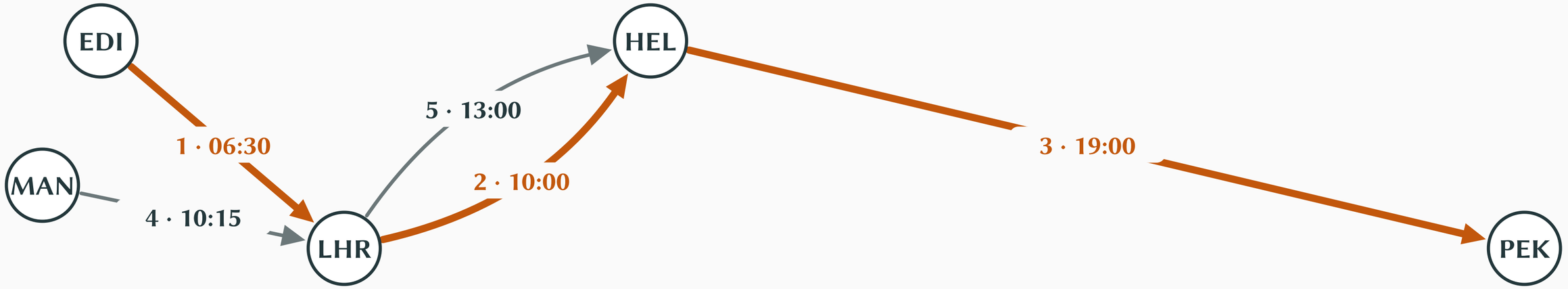
A **route** is a journey from one city to another (e.g. from Edinburgh to Beijing).

Each route may contain multiple **legs** (e.g. Edinburgh to London departing at 06:30, London to Helsinki departing at 10:00, Helsinki to Beijing departing at 19:00).

AlgoAir flies five **legs** a day.

Task: every leg flown by **exactly one** chosen route, at minimum total cost.

Flight scheduling: the routes



route	itinerary	legs	cost
R1	EDI ⟶ LHR ⟶ HEL ⟶ PEK	1, 2, 3	14
R2	MAN ⟶ LHR ⟶ HEL ⟶ PEK	3, 4, 5	13
R3	EDI ⟶ LHR ⟶ HEL	1, 5	7
R4	LHR ⟶ HEL ⟶ PEK	2, 3	12
R5	LHR ⟶ HEL	2	4
R6	MAN ⟶ LHR	4	2
R7	LHR ⟶ HEL	5	4
R8	HEL ⟶ PEK	3	9

The routes are **given**.

? *What are the variables?*

Scheduling: variables and objective

STEP 1 **Variables:** one yes/no decision per route.

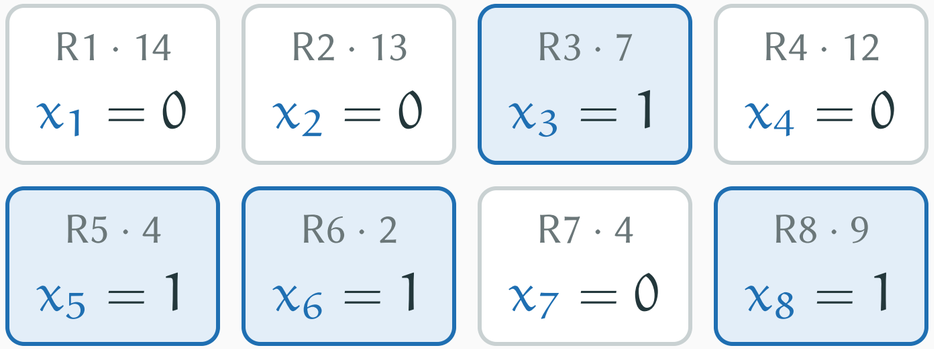
variable	meaning	values
x_j	$x_j = 1$ if route j is flown, $x_j = 0$ if it is not	0 or 1

x_j is an **indicator variable**.
The numbers a_{ij} and c_j are data.

STEP 2 **Objective:** route j costs c_j if flown, else nothing: $c_j x_j$.

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

$$= 14x_1 + 13x_2 + 7x_3 + 12x_4 + 4x_5 + 2x_6 + 4x_7 + 9x_8$$



cost $7 + 4 + 2 + 9 = 22$

Scheduling: the constraints

STEP 3 **Constraints:** every leg is flown exactly once.

$\sum_{j=1}^{n} a_{ij}\, x_j = 1$ for every leg i

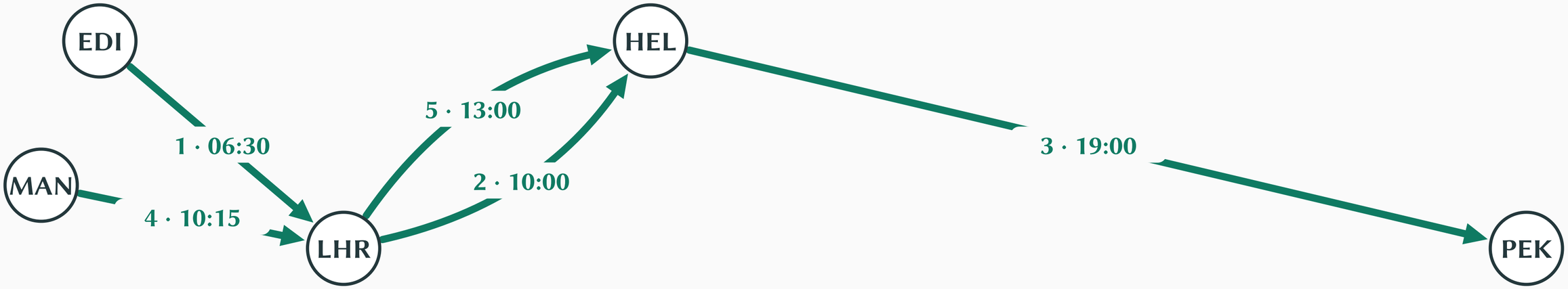
$a_{ij} = 1$ if leg i is part of route j

$$
\begin{aligned}
&\text{leg 1} && x_1 && + x_3 && && && = 1\\
&\text{leg 2} && x_1 && && + x_4 + x_5 && && = 1\\
&\text{leg 3} && x_1 + x_2 && && + x_4 && + x_8 && = 1\\
&\text{leg 4} && && x_2 && && + x_6 && = 1\\
&\text{leg 5} && && x_2 + x_3 && && + x_7 && = 1
\end{aligned}
$$

$x_j \in \{0, 1\}$ for every route j

route	itinerary	legs	cost
R1	EDI → LHR → HEL → PEK	1, 2, 3	14
R2	MAN → LHR → HEL → PEK	3, 4, 5	13
R3	EDI → LHR → HEL	1, 5	7
R4	LHR → HEL → PEK	2, 3	12
R5	LHR → HEL	2	4
R6	MAN → LHR	4	2
R7	LHR → HEL	5	4
R8	HEL → PEK	3	9

Scheduling: solved



Brute force: of the **256** sets of routes
only **3** fly every leg exactly once.

routes	cost
R1 + R6 + R7	20
R3 + R4 + R6	21
R3 + R5 + R6 + R8	22

Makespan: the problem

Jobs on parallel machines: job i takes time t_{ij} on machine j.

	M1	M2	M3
J1	2	4	1
J2	8	3	5
J3	7	4	6
J4	8	8	2
J5	6	5	8



Makespan: the time when the last machine is done.

Goal: every job on one machine, makespan as small as possible.

? *Which assignment is best?*

Makespan: variables and objective

STEP 1 **Variables:** which machine runs which job.

variable	meaning	values
x_{ij}	$x_{ij} = 1$ if job i runs on machine j	0 or 1

$nm = 15$ variables.
The times t_{ij} are data.

STEP 2 **Objective:** the **load** of machine j is

$$L_j = \sum_{i=1}^{n} t_{ij}\, x_{ij}.$$

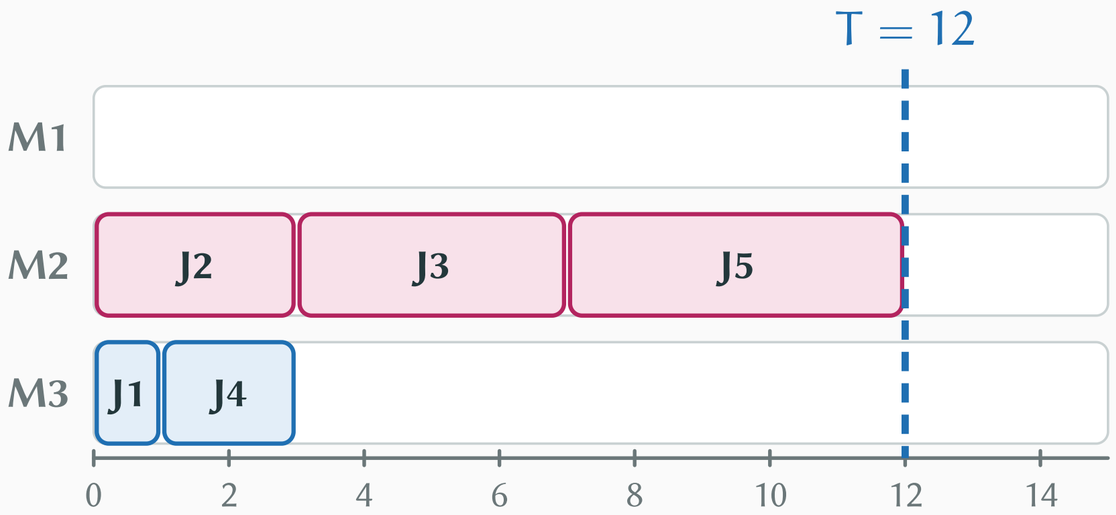
The makespan is the largest load:

$$\text{minimise} \quad \max_j \sum_{i=1}^{n} t_{ij}\, x_{ij}.$$

max is not linear: this is not an (I)LP yet.

	M1	M2	M3
J1	0	0	1
J2	0	1	0
J3	0	1	0
J4	0	0	1
J5	0	1	0
L_j	0	12	3

Makespan: a ceiling T



Minimise a maximum:
add a variable T above every load,
and minimise T.

minimise T

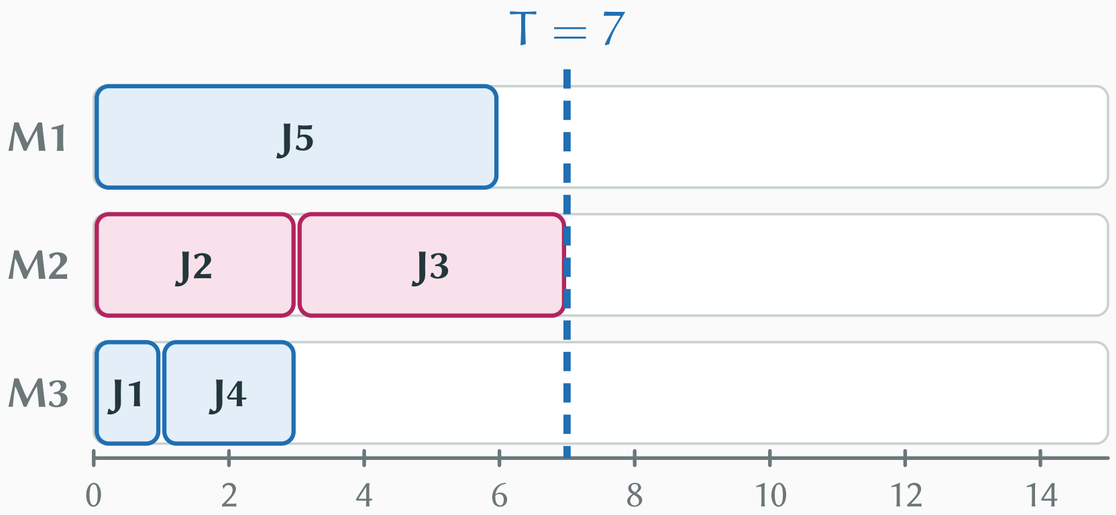
subject to

$\sum_{i=1}^{n} t_{ij}\, x_{ij} \leqslant T$ for every j — no machine works longer than T

$\sum_{j=1}^{m} x_{ij} = 1$ for every i — each job runs on exactly one machine

$x_{ij} \in \{0, 1\}$ for all i, j

Makespan: solved



Brute force over all **243** assignments.

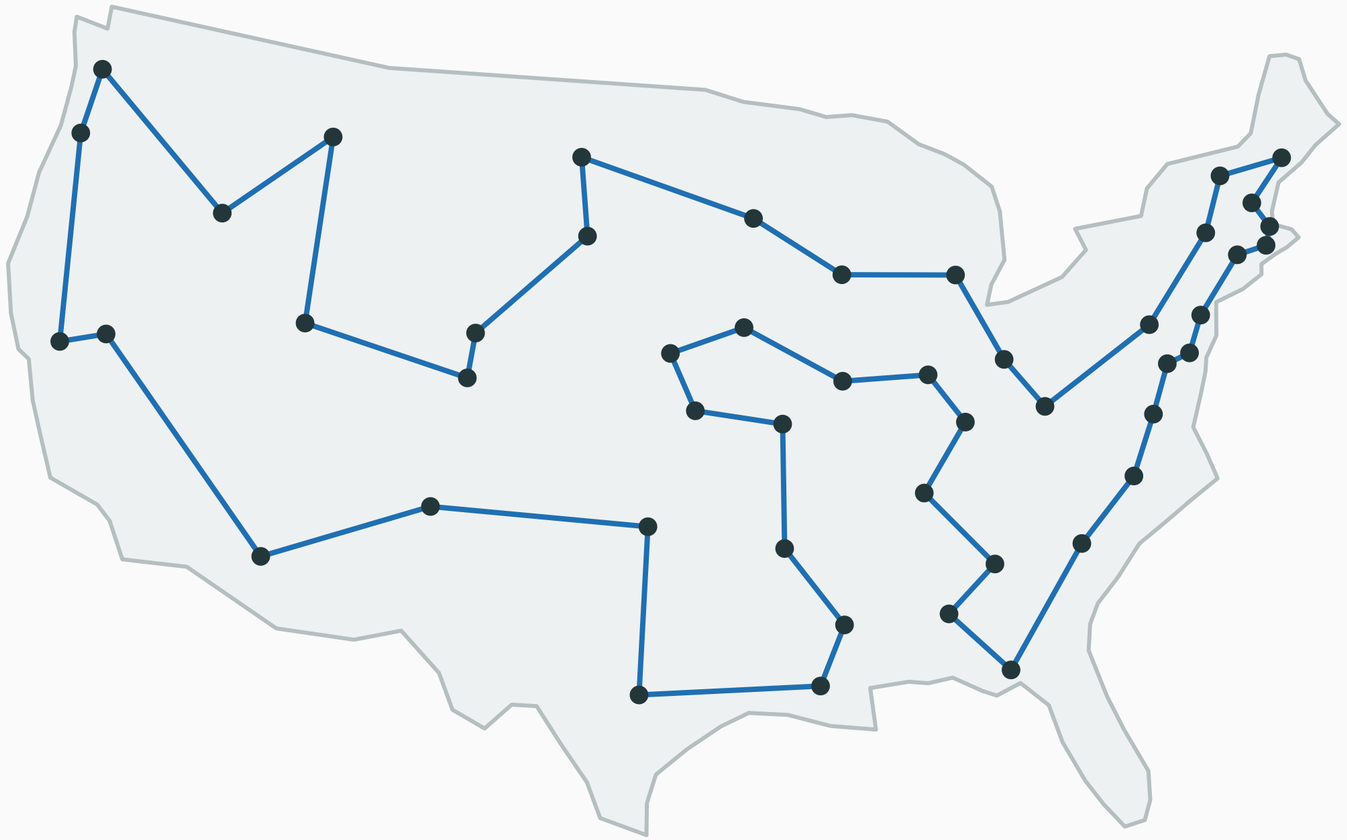
Best makespan **7**, reached by **exactly one**.

load M1: 6, M2: 7, M3: 3

Travelling salesman

How do we write a tour as an integer program?

TSP: the problem



Travelling salesman problem (TSP).

Visit the capital of every state exactly once, and return to the start.

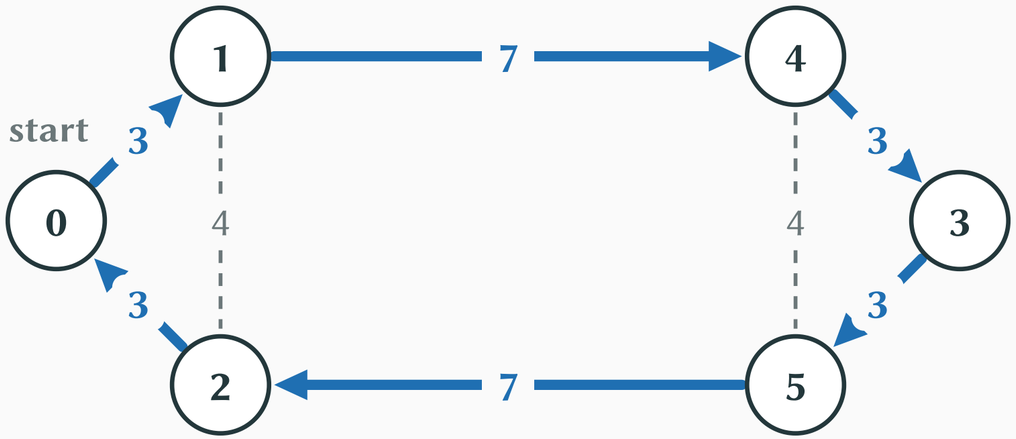
Goal: the shortest tour.

The shortest tour: about 17 100 km.

48 capitals, straight lines.

Let’s look at a smaller example.

TSP: A SMALLER EXAMPLE



shortest $3 + 7 + 3 + 3 + 7 + 3 = 26$

A smaller example: $n = 6$ cities $0, \ldots, 5$ with distances c_{ij} (in general $0, \ldots, n-1$).

Start in city 0, visit every city **exactly once**, and return to 0.

Goal: the shortest tour.

A tour is an order $0, s_1, \ldots, s_{n-1}$: $(n-1)!$ **tours**

$n = 6$: 120

$n = 48$: $2.6 \cdot 10^{59}$ (the capitals)

? *Can we model it as an ILP?*

TSP: variables and objective

STEP 1 **Variables:** which roads the tour uses.

variable	meaning	values
x_{ij}	$x_{ij} = 1$ if the tour goes directly from city i to city j	0 or 1

$i \neq j$: $n(n-1) = 30$ variables.
The distances c_{ij} are data.

STEP 2 **Objective:** we pay c_{ij} for every road used.

$$\textbf{minimise} \sum_{i=0}^{n-1} \sum_{j \neq i} c_{ij}\, x_{ij}$$

for the tour 0, 1, 2, 3, 4, 5: $3 + 4 + 9 + 3 + 4 + 9 = 32$

	0	1	2	3	4	5
0	–	**3**	3	11	9	9
1	3	–	**4**	9	7	8
2	3	4	–	**9**	8	7
3	11	9	9	–	**3**	3
4	9	7	8	3	–	**4**
5	**9**	8	7	3	4	–

distances c_{ij}
blue: $x_{ij} = 1$

TSP: degree constraints

STEP 3 **Constraints:** every city is left once and entered once.

$$\forall i,\ \sum_{j \neq i} x_{ij} = 1$$

leave city i exactly once: every row sums to 1

$$\forall j,\ \sum_{i \neq j} x_{ij} = 1$$

enter city j exactly once: every column sums to 1

> Every tour satisfies both constraints.

? *Is every x that satisfies them a tour?*

	0	1	2	3	4	5	
0	–	1	0	0	0	0	**1**
1	0	–	1	0	0	0	**1**
2	0	0	–	1	0	0	**1**
3	0	0	0	–	1	0	**1**
4	0	0	0	0	–	1	**1**
5	1	0	0	0	0	–	**1**
	1	**1**	**1**	**1**	**1**	**1**	

x_{ij} for the tour 0, 1, 2, 3, 4, 5

TSP: the catch

Two triangles, $0 \to 1 \to 2 \to 0$ and $3 \to 4 \to 5 \to 3$:
every city has one road out and one in.

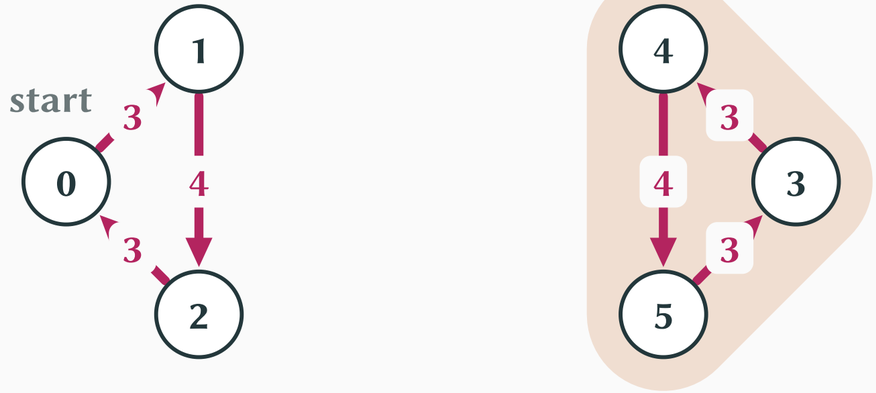
> **Degree constraints hold, but it is no tour.**
> Two **subtours**, and shorter than any tour: $20 < 26$.

Forbid it (Dantzig, Fulkerson and Johnson, 1954)**:**
for every nonempty proper subset S:

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geqslant 1$$

$2^n - 2$ sets: 62 for $n = 6$, $2.8 \cdot 10^{14}$ for $n = 48$.

? *Can we do with fewer constraints?*



length $3 + 4 + 3 + 3 + 4 + 3 = 20$

here $S = \{3, 4, 5\}$: 0 roads leave it, and $0 < 1$

TSP: positions in the tour

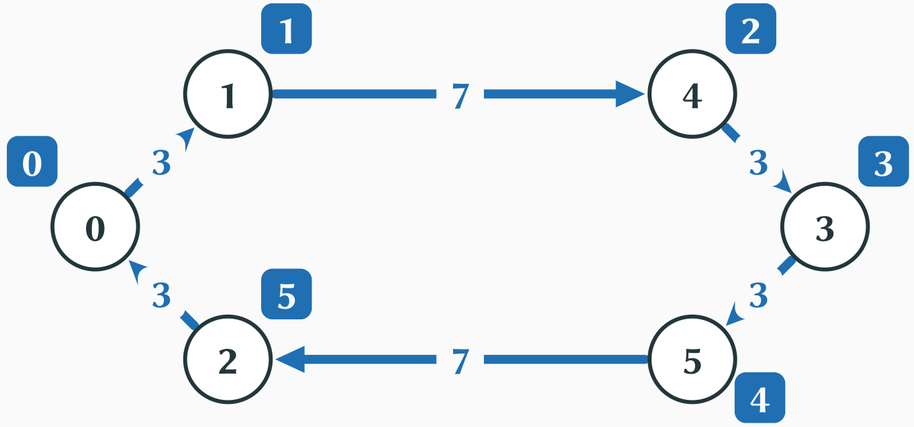
step 3 **Position variables:** t_i = position of city i.

$$t_0 = 0.$$

If the tour goes $i \to j$ ($x_{ij} = 1$):
j **comes after** i, $t_j \geqslant t_i + 1$.

If $x_{ij} = 0$: **no condition**, as weak as $t_j \geqslant 0$.

Both in one (Miller, Tucker and Zemlin, 1960).
$t_j \geqslant t_i + 1 - n\,(1 - x_{ij})$
for all i and $j \geqslant 1$ with $i \neq j$



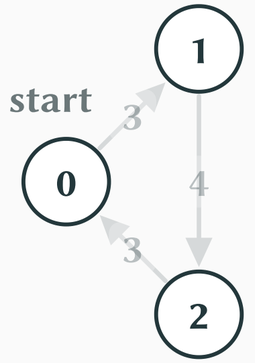
blue: the position t_i of city i

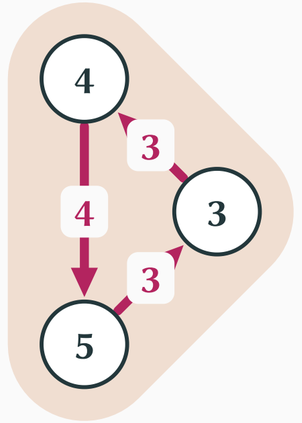
	the constraint	for our tour
$x_{14} = 1$	$t_4 \geqslant t_1 + 1$	$2 \geqslant 1 + 1 = 2$
$x_{25} = 0$	$t_5 \geqslant t_2 + 1 - 6$	$4 \geqslant 5 + 1 - 6 = 0$

TSP: no subtours

Claim. If x and t satisfy all constraints, x is a single tour.

1. The degree constraints give disjoint cycles covering all cities.
2. Suppose a cycle C avoids city 0, with r cities.
3. Each road $i \to j$ of C has $x_{ij} = 1$ and $j \geqslant 1$: $t_j - t_i \geqslant 1$.
4. Add the r inequalities: each t occurs as $+t$ and as $-t$, so they cancel: $0 \geqslant r$. Contradiction. Hence all cities lie on the cycle through city 0. □
5. Conversely, a tour with t_i = position satisfies all constraints.



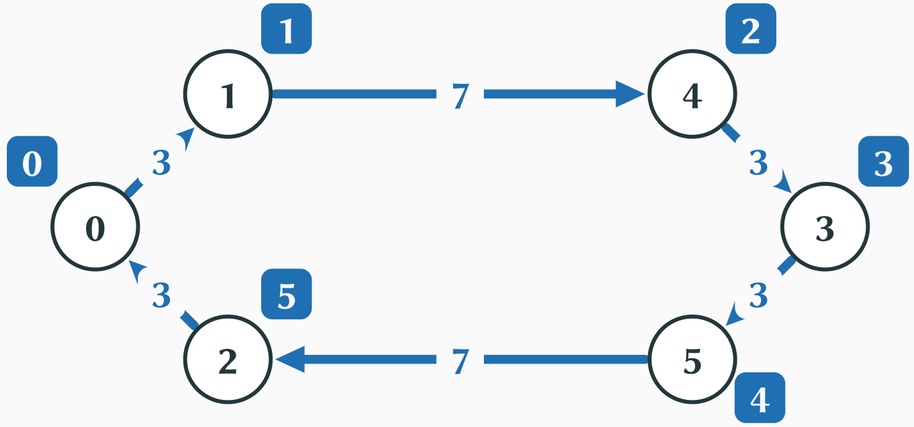


$$\begin{aligned} -t_3 + t_4 &\geqslant 1 && \text{road } 3\to4 \\ -t_4 + t_5 &\geqslant 1 && \text{road } 4\to5 \\ t_3 - t_5 &\geqslant 1 && \text{road } 5\to3 \\ \hline (-t_3 + t_4) + (-t_4 + t_5) + (t_3 - t_5) &= 0 \geqslant 3 \end{aligned}$$

impossible

TSP: the complete ILP, solved

minimise	$\sum_i \sum_{j \neq i} c_{ij}\, x_{ij}$	total length
subject to	$\sum_{j \neq i} x_{ij} = 1$ for every i	leave each city once
	$\sum_{i \neq j} x_{ij} = 1$ for every j	enter each city once
	$t_j \geqslant t_i + 1 - n(1 - x_{ij})$ for $i \geqslant 0, j \geqslant 1, i \neq j$	no subtours
	$t_0 = 0$	start in city 0
	$x_{ij} \in \{0, 1\},\ t_i \in \{0, \ldots, n-1\}$	integers



All 120 tours tried.

Shortest: 0, 1, 4, 3, 5, 2.

length $3 + 7 + 3 + 3 + 7 + 3 = 26$

$n(n-1) + (n-1)$ variables,

$n + n + (n-1)^2$ constraints.

About n^2 of each, not $(n-1)!$ tours.

Summary

1. **Three steps.** Variables, linear objective, linear constraints.
   The feasible points must be exactly the valid solutions.
2. **Yes/no decisions are 0–1 variables.**
   Scheduling: every leg flown exactly once, $\sum_j a_{ij} x_j = 1$.
3. **Maximum as a variable.**
   Makespan: minimise T subject to every load $\leqslant T$.
4. **Conditional constraints.**
   TSP: $t_j \geqslant t_i + 1 - n(1 - x_{ij})$ forbids subtours: about n^2 variables and constraints.

Reference: Vanderbei, Chapters 1.1, 23.1, 23.2