Assignment and scheduling (1/2)

7.1–7.4

Outline

The family

Minimum-cost assignment

Machines with fixed cost

Job selection

Parallel jobs

Learning objectives (1/2)

At the end of this chapter you will be able to:

- write the symbolic model of an assignment or scheduling problem from its statement;
- prove the links between machine, class and assignment variables;

Learning objectives (2/2)

At the end of this chapter you will be able to:

- model sequencing on one machine with binary precedences and a big-M computed from the data;
- close the sandwich of bounds with a constructive heuristic and a hand-built dual solution.

Prerequisites

- From this module, Chapter 6 «Links between variables»: activation, the maximum variable, the *if and only if*, big-M and precedences
- From this module, Chapter 2 «Relaxations, duality and bounds»: the bound protocol and the dual of the LP relaxation
- From the module *Combinatorial Optimization Problems*, the page «Scheduling on identical parallel machines ($P\|C_{\max}$)»: the scheduling problem $P\|C_{\max}$ and the LPT rule

The family

Introduction: The family

- Seven problems with the same skeleton: some **jobs** must be assigned to some **machines** with limited availability
- What changes from problem to problem is what is paid and what is decided
- We will learn to recognise, in different situations, the same **links between variables**: activation, maximum variable, "if and only if", big-M

What changes from one problem to another (1/2)

- The cost of the assignment (problem 1)
- The fixed cost of every machine switched on (problem 2)
- The revenue of the jobs one chooses to execute (problem 3)
- The processing time when the jobs on the same machine run in parallel (problem 4)

What changes from one problem to another (2/2)

- The setup of a class of jobs (problem 5)
- A bonus collected *if and only if* a class is complete (problem 6)
- The tardiness with respect to the due dates when the jobs follow one another on a single machine (problem 7)

The links between variables (1/2)

- **Activation** (problems 2, 3, 5): the binary variable "machine used" or "class activated" that drives the assignments, in the aggregated form

$$\sum_j t_{jm} x_{jm} \leq a_m y_m$$

  or in the disaggregated one, $x_j \leq y_c$

- **Maximum variable** (problem 4): the processing time $y_m \geq t_{jm} x_{jm}$ for every job, which at the optimum equals exactly the maximum

The links between variables (2/2)

- **If and only if** (problem 6): one direction is imposed by the constraint, the other by the objective
- **Big-M and disjunctions** (problem 7): "either j before i or i before j", with M equal to the sum of the times

The scheme of every problem

For each of the seven problems:

- model and proof of the links
- instance
- constructive heuristic: a bound from the side of feasible solutions
- dual of the LP relaxation with a hand-built dual solution: a bound from the other side
- solver solution
- additional considerations and modelling questions

Notation of the family (1/2)

Symbol	Type	Meaning
n	$\in \mathbb{Z}_{\geq 1}$	number of jobs, $j \in \{1, 2, \ldots, n\}$
k	$\in \mathbb{Z}_{\geq 1}$	number of machines, $m \in \{1, 2, \ldots, k\}$
t_{jm}	$\in \mathbb{Q}_{>0}$	processing time of job j on machine m; t_j if it does not depend on the machine
c_{jm}	$\in \mathbb{Q}_{>0}$	cost of executing job j on machine m
c_m	$\in \mathbb{Q}_{>0}$	fixed cost if machine m is used
a_m	$\in \mathbb{Q}_{>0}$	availability of machine m; a if there is a single machine

Notation of the family (2/2)

Symbol	Type	Meaning
p_m	$\in \mathbb{Z}_{\geq 1}$	maximum number of jobs on machine m
r_j	$\in \mathbb{Q}_{>0}$	revenue if job j is executed
d_j	$\in \mathbb{Q}_{>0}$	due date of job j
q	$\in \mathbb{Z}_{\geq 2}$	number of job classes, $c \in \{1, 2, \ldots, q\}$
$\mathcal{J}_c$	$\subseteq \{1, 2, \ldots, n\}$	jobs of class c (partition)
$f_c,\ s_c$	$\in \mathbb{Q}_{\geq 0}$	setup cost and setup time of class c
v_c	$\in \mathbb{Q}_{>0}$	bonus if class c is complete
u	$\in \mathbb{Q}_{>0}$	reduction of the availability if classes are mixed

Summary: The family

- The skeleton is always the same: **jobs** to be assigned to **machines** with limited availability
- What changes is the **cost** (or the revenue) and the **link** between the families of variables
- Some directions of the links are imposed by the constraints, others follow from **optimality**
- For every problem we build two bounds: a **primal** one (heuristic) and a **dual** one (LP relaxation)

Minimum-cost assignment

Introduction: Minimum-cost assignment

- The first problem of the family: every job must be processed by a machine, and every machine has a limited available time
- It is the **generalised assignment** problem: an assignment with a knapsack constraint per machine
- There are no links to prove: there is a single family of variables. It serves to fix the scheme of the two bounds we will use in every problem

Problem 7.1: minimum-cost assignment (1/2)

- A company has to process $n \in \mathbb{Z}_{\geq 1}$ jobs on $k \in \mathbb{Z}_{\geq 1}$ machines
- For each job j and each machine m, the value $t_{jm} \in \mathbb{Q}_{>0}$ is the processing time in minutes and the value $c_{jm} \in \mathbb{Q}_{>0}$ is the cost in euros of processing job j on machine m

Problem 7.1: minimum-cost assignment (2/2)

- For each machine m, the value $a_m \in \mathbb{Q}_{>0}$ is the available processing time in minutes
- Each machine handles one job at a time (no parallel processing)
- The company seeks a **minimum-cost** assignment of all jobs to the machines

The problem in words

- **We decide** on which machine each job is processed
- **The objective**: minimum total cost
- **The constraints**: every job is processed by exactly one machine; the total time of the jobs assigned to a machine does not exceed its availability
- It is the **generalised assignment** problem: an assignment with a knapsack constraint per machine

Data (input of the model)

Symbol	Type	Meaning
n	$\in \mathbb{Z}_{\geq 1}$	number of jobs, $j \in \{1, 2, \ldots, n\}$
k	$\in \mathbb{Z}_{\geq 1}$	number of machines, $m \in \{1, 2, \ldots, k\}$
t_{jm}	$\in \mathbb{Q}_{>0}$	processing time of job j on machine m
c_{jm}	$\in \mathbb{Q}_{>0}$	cost of processing job j on machine m
a_m	$\in \mathbb{Q}_{>0}$	availability of machine m

Decision variables

We introduce the following $n\,k$ binary variables:

$$x_{jm} = \begin{cases} 1 & \text{if job } j \text{ is processed by machine } m, \\ 0 & \text{otherwise,} \end{cases}$$

$$\forall j \in \{1, 2, \ldots, n\},\ \forall m \in \{1, 2, \ldots, k\}.$$

The model

Model 7.1 — minimum-cost assignment

$$\min \quad \sum_{j=1}^{n} \sum_{m=1}^{k} c_{jm}\, x_{jm} \tag{7.1a}$$

$$\text{subject to} \quad \sum_{m=1}^{k} x_{jm} = 1, \qquad \forall j \in \{1, 2, \ldots, n\}, \tag{7.1b}$$

$$\sum_{j=1}^{n} t_{jm}\, x_{jm} \leq a_m, \qquad \forall m \in \{1, 2, \ldots, k\}, \tag{7.1c}$$

$$x_{jm} \in \{0, 1\}, \quad \forall j \in \{1, 2, \ldots, n\}, \ \forall m \in \{1, 2, \ldots, k\}. \tag{7.1d}$$

Description of the objective function and constraints

- The objective function minimises the total processing cost, the sum of the costs of the chosen assignments
- Constraints (7.1b) ensure that each job is assigned to exactly one machine, so that all jobs are processed (n constraints)
- Constraints (7.1c) guarantee that the total processing time of the jobs assigned to each machine does not exceed its availability (k constraints)
- The last constraints define the variables of the model

No link to prove

- The model has a single family of variables: there are no links to prove
- Constraints (7.1c) are of the "capacity/resource" kind: two quantities of the same nature (minutes required and minutes available) compared, no logical implication to write
- The two bounds we build — the **primal** one, from the heuristic, and the **dual** one, from the relaxation — are the two sides of the sandwich that a MIP solver tries to close

The model in gurobipy

```
m = gp.Model("assignment")
x = m.addVars(n, k, vtype=GRB.BINARY, name="x")
m.setObjective(gp.quicksum(c[j][mm] * x[j, mm]
    for j in range(n) for mm in range(k)), GRB.
        MINIMIZE)
m.addConstrs((x.sum(j, "*") == 1 for j in range(n)),
             name="assign")
m.addConstrs((gp.quicksum(t[j][mm] * x[j, mm]
    for j in range(n)) <= a[mm] for mm in range(k)),
    name="availability")
m.optimize()
```

The instance

$n = 3$ jobs, $k = 3$ machines:

t_{jm}	$m = 1$	$m = 2$	$m = 3$
$j = 1$	2	1	3
$j = 2$	3	4	2
$j = 3$	4	5	3

c_{jm}	$m = 1$	$m = 2$	$m = 3$
$j = 1$	5	10	2
$j = 2$	5	4	6
$j = 3$	5	4	6

	$m = 1$	$m = 2$	$m = 3$
a_m	5	6	7

The model for the instance

The instance model

$$
\begin{aligned}
\min\ & 5x_{11} + 10x_{12} + 2x_{13} + 5x_{21} + 4x_{22} + 6x_{23} + 5x_{31} + 4x_{32} + 6x_{33} \\
\text{subject to}\ & x_{11} + x_{12} + x_{13} = 1 \\
& x_{21} + x_{22} + x_{23} = 1 \\
& x_{31} + x_{32} + x_{33} = 1 \\
& 2x_{11} + 3x_{21} + 4x_{31} \leq 5 \\
& x_{12} + 4x_{22} + 5x_{32} \leq 6 \\
& 3x_{13} + 2x_{23} + 3x_{33} \leq 7 \\
& x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33} \in \{0, 1\}
\end{aligned}
$$

Constructive heuristic: the primal bound

- Three heuristics inspired by bin packing, all linear in the number of job–machine pairs
- **Next-fit**: one machine is loaded at a time and the next one is opened when a job no longer fits
- **First-fit**: each job goes on the first machine with enough residual availability
- **Best-fit**: among the machines with enough availability, the one with minimum cost is chosen — the most natural here, since the objective is a cost

Run on the instance (1/2)

- **Step 1.** Job 1: residual availabilities $ra = (5, 6, 7)$; every machine is enough; costs $5, 10, 2$: the minimum is machine 3, hence $x[1][3] = 1$ and $ra[3] = 7 - 3 = 4$
- **Step 2.** Job 2: $ra = (5, 6, 4)$; every machine is enough; costs $5, 4, 6$: the minimum is machine 2, hence $x[2][2] = 1$ and $ra[2] = 6 - 4 = 2$
- **Step 3.** Job 3: $ra = (5, 2, 4)$; machine 2 is not enough $(5 > 2)$; among the others, costs 5 and 6: the minimum is machine 1, hence $x[3][1] = 1$

Run on the instance (2/2)

The feasible solution found is

$$\bar{x}_{13} = 1, \quad \bar{x}_{22} = 1, \quad \bar{x}_{31} = 1,$$

all the other $\bar{x}_{jm} = 0$, of value $2 + 4 + 5 = 11$. Hence

$$z(MILP) \leq UB = 11.$$

- Next-fit and first-fit both find $x_{11} = x_{21} = x_{32} = 1$, of value 14: a worse bound

LP relaxation and dual

- The LP relaxation replaces $x_{jm} \in \{0, 1\}$ with $x_{jm} \geq 0$ (the constraint $x_{jm} \leq 1$ is implied by constraints (1))
- A free dual variable μ_j for each constraint (1), a non-positive one π_m for each constraint (2)

Dual of the LP relaxation of model 7.1

$$\max \quad \sum_{j=1}^{n} \mu_j + \sum_{m=1}^{k} a_m \pi_m \tag{7.3a}$$

$$\text{subject to} \quad \mu_j + t_{jm} \pi_m \leq c_{jm}, \quad \forall j \in \{1, 2, \ldots, n\}, \ \forall m \in \{1, 2, \ldots, k\}, \tag{7.3b}$$

$$\mu_j \gtreqless 0, \qquad \forall j \in \{1, 2, \ldots, n\}, \tag{7.3c}$$

$$\pi_m \leq 0, \qquad \forall m \in \{1, 2, \ldots, k\}. \tag{7.3d}$$

Description of the dual

- μ_j is the value attached to job j, which has to be assigned anyway; π_m the (non-positive) price of one unit of capacity of machine m
- The objective prices the jobs and the available capacity
- Constraints (7.3b) are the columns of the x_{jm}: putting job j on machine m is worth μ_j and consumes t_{jm} of capacity at price π_m; the balance cannot exceed the cost c_{jm}
- The μ_j are free (equality constraints), the π_m non-positive ($\leq$ constraints in a minimisation)

The dual of the instance

The instance model

$$
\begin{array}{rrrrrrrrrrrrr}
\max & \mu_1 & + & \mu_2 & + & \mu_3 & + 5\pi_1 & + 6\pi_2 & + 7\pi_3 & & \\
\text{subject to} & \mu_1 & & & & & + 2\pi_1 & & & \leq & 5 \\
& \mu_1 & & & & & & + \pi_2 & & \leq & 10 \\
& \mu_1 & & & & & & & + 3\pi_3 & \leq & 2 \\
& & & \mu_2 & & & + 3\pi_1 & & & \leq & 5 \\
& & & \mu_2 & & & & + 4\pi_2 & & \leq & 4 \\
& & & \mu_2 & & & & & + 2\pi_3 & \leq & 6 \\
& & & & & \mu_3 & + 4\pi_1 & & & \leq & 5 \\
& & & & & \mu_3 & & + 5\pi_2 & & \leq & 4 \\
& & & & & \mu_3 & & & + 3\pi_3 & \leq & 6 \\
& \mu_1, & & \mu_2, & & \mu_3 & & & & \geq & 0 \\
& & & & & & \pi_1, & \pi_2, & \pi_3 & \leq & 0
\end{array}
$$

A hand-built dual solution

- With $\bar{\pi}_m = 0$ for every machine, constraints (7.3b) become $\mu_j \leq c_{jm}$ for every m: the largest feasible value is

$$\bar{\mu}_1 = \min\{5, 10, 2\} = 2, \quad \bar{\mu}_2 = \min\{5, 4, 6\} = 4, \quad \bar{\mu}_3 = 4,$$

  with value $2 + 4 + 4 + 0 = 10$
- By weak duality

$$LB = 10 \;\leq\; z(LP) \;\leq\; z(MILP) \;\leq\; UB = 11$$

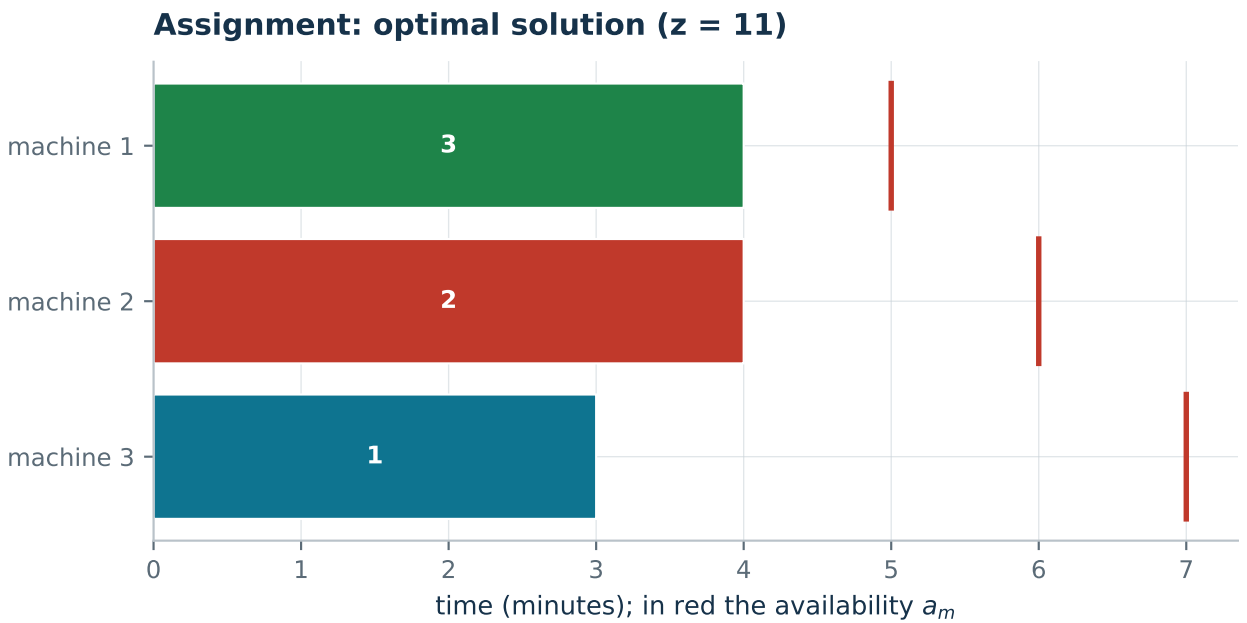
- The recipe has a meaning: "every job costs at least its minimum cost" is a lower bound anyone would write down; the dual formalises it

What the solver says (1/2)

- The optimum of the LP relaxation is $z(LP) = 53/5 = 10.6$, equal to the optimum of the dual (strong duality), with $\tilde{\mu} = (2,\ 4.8,\ 5)$ and $\tilde{\pi} = (0,\ -0.2,\ 0)$: machine 2 is the tight resource
- The MILP has optimum $z(MILP) = 11$ with $\tilde{x}_{13} = \tilde{x}_{22} = \tilde{x}_{31} = 1$
- The best-fit heuristic had found the optimal solution, but only the solver certifies it, because the hand-built dual bound stopped at 10
- No integer between 10.6 and 11: $\lceil 53/5 \rceil = 11$ closes the gap — an argument that works because the costs are integer

What the solver says (2/2)

UB (best-fit)	LB (hand dual)	$z(LP)$	$z(MILP)$	heuristic gap
11	10	53/5	11	0.0%



Additional considerations

- The $n\,k$ inequalities $x_{jm} \leq 1$ are valid for the model, but they are implied by constraints (1) and do not strengthen the relaxation (indeed $z(LP) = z(LP^+)$)
- If a job j does not fit on a machine m ($t_{jm} > a_m$), the variable x_{jm} can be fixed to zero before solving: constraint (2) would force it anyway, but the model is smaller and the relaxation does not get worse

Exercise: fixed cost per used machine

- Every machine that processes at least one job costs an extra $g_m = 3$ euros to switch on
- Model the fixed cost and find the new optimum
- Which link of the chapter "Links between variables" comes into play?

The final comparison (1/2)

The three numbers that every problem of the course produces:

	value	what it is
UB	11	heuristic solution
LB	10	certified bound
$z(MILP)$	11	MILP optimum

The final comparison (2/2)

- Gap of the heuristic with respect to the optimum:

$$\frac{|UB - z(MILP)|}{|z(MILP)|} = 0.0\%$$

- Certified gap: the one known *before* solving the MILP, which therefore cannot have it as its denominator
- In absolute terms $UB - LB = 1$; as a percentage, against the heuristic solution,

$$\frac{UB - LB}{|UB|} = 9.1\%$$

A worked variant: jobs 1 and 3 on the same machine (1/2)

- Jobs 1 and 3 use the same tool and must be processed by the same machine
- It is a link between two variables of the same family: for every machine m, $x_{1m} = 1$ if and only if $x_{3m} = 1$, that is

$$x_{1m} = x_{3m}, \qquad \forall m \in \{1, 2, \ldots, k\}$$

(k linear constraints)

- Both directions are imposed by the constraint; the assignment constraints then guarantee that the common machine is unique

A worked variant: jobs 1 and 3 on the same machine (2/2)

- The pair must go where $t_{1m} + t_{3m} \leq a_m$: not on machine 1 ($2 + 4 > 5$), yes on machine 2 (cost $10 + 4$), yes on machine 3 (cost $2 + 6$)
- With job 2 on machine 2 (cost 4), the new optimum is $2 + 6 + 4 = 12$

	value	what it is
UB	12	heuristic solution
LB	12	hand-built dual certificate
$z(MILP)$	12	MILP optimum

The two bounds meet: the problem closes without the solver.

Summary: Minimum-cost assignment

- A single family of binary variables x_{jm}: no link to prove
- Assignment constraints ($= 1$) and capacity constraints ($\leq a_m$), one knapsack per machine
- Minimum-cost best-fit heuristic: $UB = 11$; hand-built dual with $\pi = 0$: $LB = 10$
- $z(LP) = 53/5$, $z(MILP) = 11$: the heuristic was already optimal, but only the solver certifies it

Machines with fixed cost

Introduction: Machines with fixed cost

- Compared with problem 7.1 the cost is no longer on the assignments, but on the **machines switched on**
- **Activation** variables are needed: one binary per machine, saying whether the machine is used
- We will see the activation link in its **aggregated** form, and the difference between a direction imposed by the constraints and one imposed by optimality

Problem 7.2: machines with fixed cost (1/2)

- A company needs to process $n \in \mathbb{Z}_{\geq 1}$ jobs and has $k \in \mathbb{Z}_{\geq 1}$ machines
- For each job j and each machine m, the value $t_{jm} \in \mathbb{Q}_{>0}$ is the processing time in minutes of job j on machine m
- For each machine m, the value $a_m \in \mathbb{Q}_{>0}$ is the availability in minutes and the value $c_m \in \mathbb{Q}_{>0}$ is the cost in euros if the machine is used

Problem 7.2: machines with fixed cost (2/2)

- Each machine handles one job at a time
- The company wants to assign all jobs to the machines **minimising the cost of the machines used**

The problem in words

- **We decide** which machines to switch on and on which machine each job is processed
- **The objective**: total cost of the machines switched on
- **The constraints**: every job on exactly one machine; a machine switched off processes no job; a machine switched on does not exceed its availability
- Compared with problem 7.1 the cost moves from the assignments to the machines: **activation** variables are needed

Data (input of the model)

Symbol	Type	Meaning
n	$\in \mathbb{Z}_{\geq 1}$	number of jobs, $j \in \{1, 2, \ldots, n\}$
k	$\in \mathbb{Z}_{\geq 1}$	number of machines, $m \in \{1, 2, \ldots, k\}$
t_{jm}	$\in \mathbb{Q}_{>0}$	processing time of job j on machine m
a_m	$\in \mathbb{Q}_{>0}$	availability of machine m
c_m	$\in \mathbb{Q}_{>0}$	fixed cost if machine m is used

Decision variables

We introduce the following $n\,k + k$ binary variables:

$$\begin{cases} x_{jm} = 1 \text{ if job } j \text{ is processed by machine } m,\ 0 \text{ otherwise,} \\ y_m = 1 \text{ if machine } m \text{ is used},\ 0 \text{ otherwise,} \end{cases}$$

$$\forall j \in \{1, 2, \ldots, n\},\ \forall m \in \{1, 2, \ldots, k\}.$$

The model

Model 7.2 — machines with fixed cost

$$\min \quad \sum_{m=1}^{k} c_m y_m \tag{7.4a}$$

$$\text{subject to} \quad \sum_{m=1}^{k} x_{jm} = 1, \qquad \forall j \in \{1, 2, \ldots, n\}, \tag{7.4b}$$

$$-\sum_{j=1}^{n} t_{jm} x_{jm} + a_m y_m \geq 0, \qquad \forall m \in \{1, 2, \ldots, k\}, \tag{7.4c}$$

$$x_{jm} \in \{0, 1\}, \quad \forall j \in \{1, 2, \ldots, n\}, \ \forall m \in \{1, 2, \ldots, k\}, \tag{7.4d}$$

$$y_m \in \{0, 1\}, \qquad \forall m \in \{1, 2, \ldots, k\}. \tag{7.4e}$$

Description of the objective function and constraints

- The objective function minimises the total cost of the machines used
- Constraints (7.4b) ensure that each job is assigned to exactly one machine (n constraints)
- Constraints (7.4c) link the assignment variables with the usage variables and impose the capacity restrictions: if at least one job is assigned, the machine is used; no job is assigned to an unused machine; a machine in use does not exceed its availability (k constraints)
- The last constraints define the variables of the model

The link between the variables (1/3)

- For each machine m, if the total processing time of the assigned jobs is positive, then the machine must be used:

$$\sum_{j=1}^{n} t_{jm} x_{jm} > 0 \implies y_m = 1$$

- By contraposition:

$$y_m = 0 \implies \sum_{j=1}^{n} t_{jm} x_{jm} = 0$$

that is, since $t_{jm} > 0$, no job can be assigned to an unused machine

The link between the variables (2/3)

This direction is imposed by constraints (7.4c), which give $\sum_j t_{jm} x_{jm} \leq a_m y_m$:

- if $\sum_j t_{jm} x_{jm} > 0$, then $a_m y_m > 0$, hence $y_m > 0$ and, being binary, $y_m = 1$
- if $y_m = 0$, the constraint gives $\sum_j t_{jm} x_{jm} \leq 0$ and, since $t_{jm} > 0$ and $x_{jm} \geq 0$, this forces $x_{jm} = 0$ for every j

The link between the variables (3/3)

- The opposite direction

$$y_m = 1 \implies \sum_{j=1}^{n} t_{jm} x_{jm} > 0$$

  says that a used machine has at least one job

- It is *not* imposed by the constraints: $y_m = 1$ with no jobs is feasible
- It follows from the objective in every optimum: since $c_m > 0$, setting $y_m = 0$ keeps the constraints (constraint (7.4c) becomes $0 \geq 0$) and decreases the objective by $c_m > 0$
- Therefore in every optimal solution a machine with no jobs is not used

The model in gurobipy

```
m = gp.Model("fixed_cost")
x = m.addVars(n, k, vtype=GRB.BINARY, name="x")
y = m.addVars(k, vtype=GRB.BINARY, name="y")
m.setObjective(gp.quicksum(c[mm] * y[mm]
    for mm in range(k)), GRB.MINIMIZE)
m.addConstrs((x.sum(j, "*") == 1 for j in range(n)),
             name="assign")
m.addConstrs((-gp.quicksum(t[j][mm] * x[j, mm]
    for j in range(n)) + a[mm] * y[mm] >= 0
    for mm in range(k)), name="link")
m.optimize()
```

The instance

$n = 3$ jobs, $k = 3$ machines:

t_{jm}	$m = 1$	$m = 2$	$m = 3$
$j = 1$	6	5	3
$j = 2$	5	10	2
$j = 3$	20	13	10

	$m = 1$	$m = 2$	$m = 3$
c_m	8	7	5
a_m	25	20	12

The model for the instance

Model 7.2 on the instance

$$
\begin{aligned}
\min \quad & 8y_1 + 7y_2 + 5y_3 \\
\text{subject to} \quad & x_{11} + x_{12} + x_{13} = 1 \\
& x_{21} + x_{22} + x_{23} = 1 \\
& x_{31} + x_{32} + x_{33} = 1 \\
& -6x_{11} - 5x_{21} - 20x_{31} + 25y_1 \geq 0 \\
& -5x_{12} - 10x_{22} - 13x_{32} + 20y_2 \geq 0 \\
& -3x_{13} - 2x_{23} - 10x_{33} + 12y_3 \geq 0 \\
& x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33} \in \{0,1\} \\
& y_1, y_2, y_3 \in \{0,1\}
\end{aligned}
$$

Constructive heuristic: the primal bound

- The same three families as in problem 7.1, with the best-fit criterion becoming the **minimum time**
- There are no assignment costs: it pays to consume little availability
- It is a constructive heuristic: a single solution is built, one element at a time, never backtracking

Run on the instance (1/2)

- **Step 1.** Job 1: $ra = (25, 20, 12)$; times $6, 5, 3$, all feasible; minimum on machine 3: $x[1][3] = 1$, $ra[3] = 12 - 3 = 9$
- **Step 2.** Job 2: $ra = (25, 20, 9)$; times $5, 10, 2$; minimum on machine 3: $x[2][3] = 1$, $ra[3] = 9 - 2 = 7$
- **Step 3.** Job 3: $ra = (25, 20, 7)$; machine 3 is not enough $(10 > 7)$; between 20 and 13 the minimum is machine 2: $x[3][2] = 1$, $ra[2] = 20 - 13 = 7$

Run on the instance (2/2)

The machines used are 2 and 3:

$$\bar{y} = (0, 1, 1), \qquad \text{value } 7 + 5 = 12, \qquad z(MILP) \leq UB = 12.$$

- Next-fit and first-fit use machines 1 and 2 (value 15)
- The "opened machines first" variants also give 15
- The minimum-time best-fit is the only one of the five to find 12

LP relaxation and dual

- We relax to $x_{jm} \geq 0$ and $y_m \geq 0$; a free dual μ_j for constraints (1), a non-negative π_m for constraints (2) ($\geq$ in a minimisation)

Dual of the LP relaxation of model 7.2

$$\max \quad \sum_{j=1}^{n} \mu_j \tag{7.5a}$$

$$\text{subject to} \quad \mu_j - t_{jm}\,\pi_m \leq 0, \quad \forall j \in \{1,2,\ldots,n\},\ \forall m \in \{1,2,\ldots,k\}, \tag{7.5b}$$

$$a_m\,\pi_m \leq c_m, \quad \forall m \in \{1,2,\ldots,k\}, \tag{7.5c}$$

$$\mu_j \gtreqless 0, \quad \forall j \in \{1,2,\ldots,n\}, \tag{7.5d}$$

$$\pi_m \geq 0, \quad \forall m \in \{1,2,\ldots,k\}. \tag{7.5e}$$

Description of the dual

- μ_j is the value of job j, π_m the price of one unit of capacity of machine m
- The objective prices the jobs only: capacity by itself is not paid for
- Constraints (7.5b) are the columns of the x_{jm}: assigning job j to machine m collects μ_j and consumes t_{jm} of capacity at price π_m; the balance cannot be positive, because the x_{jm} do not appear in the primal objective
- Constraints (7.5c) are the columns of the y_m: all the capacity a_m made available, priced at π_m, cannot be worth more than the fixed cost c_m

The dual of the instance

The instance model

$$
\begin{array}{rrrrrrrrrrr}
\max & \mu_1 & + \mu_2 & + \mu_3 & & & & & & & \\
\text{subject to} & \mu_1 & & & - 6\pi_1 & & & & \leq & 0 \\
& \mu_1 & & & & - 5\pi_2 & & & \leq & 0 \\
& \mu_1 & & & & & - 3\pi_3 & & \leq & 0 \\
& & \mu_2 & & - 5\pi_1 & & & & \leq & 0 \\
& & \mu_2 & & & - 10\pi_2 & & & \leq & 0 \\
& & \mu_2 & & & & - 2\pi_3 & & \leq & 0 \\
& & & \mu_3 & - 20\pi_1 & & & & \leq & 0 \\
& & & \mu_3 & & - 13\pi_2 & & & \leq & 0 \\
& & & \mu_3 & & & - 10\pi_3 & & \leq & 0 \\
& & & & 25\pi_1 & & & & \leq & 8 \\
& & & & & 20\pi_2 & & & \leq & 7 \\
& & & & & & 12\pi_3 & & \leq & 5 \\
& \mu_1, & \mu_2, & \mu_3 & & & & & \gtreqless & 0 \\
& & & & \pi_1, & \pi_2, & \pi_3 & & \geq & 0
\end{array}
$$

A hand-built dual solution (1/2)

- Constraints (7.5c) say how large each π_m can be at most: take the maximum, $\bar{\pi}_m = c_m/a_m$ (the cost per minute of each machine)

$$\bar{\pi}_1 = \frac{8}{25}, \qquad \bar{\pi}_2 = \frac{7}{20}, \qquad \bar{\pi}_3 = \frac{5}{12}$$

- Constraints (7.5b) then give $\mu_j \leq t_{jm}\bar{\pi}_m$ for every m, and the largest feasible value is

$$\bar{\mu}_1 = \min\left\{\frac{48}{25}, \frac{7}{4}, \frac{5}{4}\right\} = \frac{5}{4}, \qquad \bar{\mu}_2 = \min\left\{\frac{8}{5}, \frac{7}{2}, \frac{5}{6}\right\} = \frac{5}{6},$$
$$\bar{\mu}_3 = \min\left\{\frac{32}{5}, \frac{91}{20}, \frac{25}{6}\right\} = \frac{25}{6}$$

A hand-built dual solution (2/2)

- The value of this feasible dual solution is

$$\frac{5}{4} + \frac{5}{6} + \frac{25}{6} = \frac{25}{4}$$

- By weak duality

$$LB = \frac{25}{4} \leq z(MILP) \leq UB = 12$$

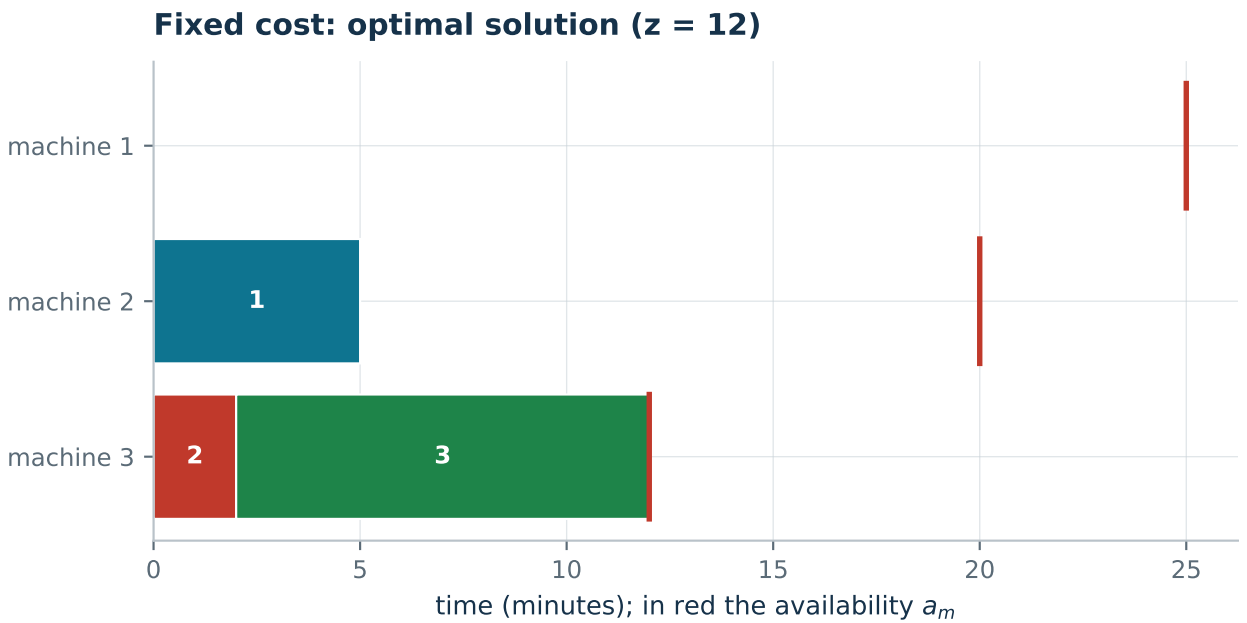
- The reading: every job "pays" at least the cost per minute of the machine that is cheapest for it, times its time
- It is a weak bound: the fixed cost is paid in full as soon as the machine is used, but the relaxation spreads it over the minutes

What the solver says (1/2)

- The optimum of the relaxation without the bounds is exactly $z(LP) = 25/4$: the hand-built solution is optimal for the dual
- With the bounds $y_m \leq 1$ and $x_{jm} \leq 1$ we get $z(LP^+) = 1273/200 = 6.365$; adding the disaggregated constraints $x_{jm} \leq y_m$ it rises to $440/67 = 6.567$
- The integer optimum is $z(MILP) = 12$, with machines 2 and 3 on and $\tilde{x}_{12} = \tilde{x}_{23} = \tilde{x}_{33} = 1$
- The gap between 6.25 and 12 measures how far the relaxation is from the integer: "fractional" fixed costs fool the LP

What the solver says (2/2)

UB (best-fit)	LB (hand dual)	$z(LP)$	$z(LP^+)$	$z(MILP)$
12	25/4	25/4	1273/200	12



Additional considerations: the disaggregated form (1/2)

- The inequalities $y_m \leq 1$ (k) and $x_{jm} \leq 1$ $(n\,k)$ are valid; the latter are implied by constraints (1), the former are not: indeed they strengthen the relaxation ($6.25 \rightarrow 6.365$)
- "If at least one job is assigned to m then m is used" is the Boolean implication

$$(x_{1m} \quad \texttt{OR} \quad x_{2m} \quad \texttt{OR} \quad \cdots \quad \texttt{OR} \quad x_{nm}) \Longrightarrow y_m$$

whose contrapositive is

$$\texttt{NOT}\ y_m \Longrightarrow (\texttt{NOT}\ x_{1m} \quad \texttt{AND} \quad \cdots \quad \texttt{AND} \quad \texttt{NOT}\ x_{nm})$$

Additional considerations: the disaggregated form (2/2)

- By De Morgan and distributivity one gets the CNF

$$(\texttt{NOT}\ x_{1m}\ \ \texttt{OR}\ \ y_m)\ \ \texttt{AND}\ \ \cdots\ \ \texttt{AND}\ \ (\texttt{NOT}\ x_{nm}\ \ \texttt{OR}\ \ y_m)$$

  i.e. the constraints $1 - x_{jm} + y_m \geq 1$:

$$x_{jm} \leq y_m, \qquad \forall j,\ \forall m$$

  ($n\,k$ valid inequalities, the **disaggregated** form of the link)

- The model implies them, but the relaxation does not: added, they bring $z(LP^+)$ to $440/67$ — same set of integer solutions, tighter relaxation

Additional considerations: the opposite direction

- "If no job is assigned then the machine is not used" is the clause

$$\texttt{NOT}\ y_m \quad \texttt{OR} \quad x_{1m} \quad \texttt{OR} \quad \cdots \quad \texttt{OR} \quad x_{nm}$$

  that is

$$\sum_{j=1}^{n} x_{jm} \geq y_m, \qquad \forall m \in \{1, 2, \ldots, k\}$$

- It is not a valid inequality (the model allows $y_m = 1$ with no jobs), but it can be added without losing the optimum, because $c_m > 0$

Exercise: minimum usage of a machine switched on

- Switching a machine on for a few minutes makes no sense: every machine used must work at least $\ell = 8$ minutes
- Model the constraint
- Find the new optimum for the instance

The final comparison (1/2)

The three numbers that every problem of the course produces:

	value	what it is
UB	12	heuristic solution
LB	$\frac{25}{4}$	certified bound
$z(MILP)$	12	MILP optimum

The final comparison (2/2)

- Gap of the heuristic with respect to the optimum:

$$\frac{|UB - z(MILP)|}{|z(MILP)|} = 0.0\%$$

- Certified gap: the one known *before* solving the MILP
- In absolute terms $UB - LB = \frac{23}{4}$; as a percentage, against the heuristic solution,

$$\frac{UB - LB}{|UB|} = 47.9\%$$

A worked variant: link between two activations (1/2)

- Machine 3 shares the power supply with machine 1: if machine 1 is used, machine 3 must be used too
- It is an implication between two binaries, $y_1 \implies y_3$, i.e. `NOT` y_1 `OR` y_3, already in CNF:

$$1 - y_1 + y_3 \geq 1 \quad \iff \quad y_1 \leq y_3$$

- It imposes that $y_1 = 1$ forces $y_3 = 1$ and, by contraposition, that $y_3 = 0$ forces $y_1 = 0$; it does not impose the converse

A worked variant: link between two activations (2/2)

- On the instance the constraint does not change the optimum, 12, because the optimal solution does not use machine 1
- A link between *activations* does not touch the relaxation, which can switch on half a machine: what grows is the integer optimum

	value	what it is
UB	12	heuristic solution
LB	$\frac{25}{4}$	hand-built dual certificate
$z(LP)$	$\frac{25}{4}$	relaxation without the bounds
$z(LP^+)$	$\frac{1273}{200}$	relaxation with the bounds
$z(MILP)$	12	MILP optimum

Summary: Machines with fixed cost

- Assignment variables x_{jm} and **activation** variables y_m
- $a_m y_m \geq \sum_j t_{jm} x_{jm}$: a single constraint acts as **aggregated activation** and as **capacity**
- The direction "machine on $\Rightarrow$ used" follows from optimality, because $c_m > 0$
- The disaggregated form $x_{jm} \leq y_m$ does not change the integer solutions but tightens the relaxation
- Bounds: $LB = 25/4$ (hand dual), $UB = 12$ (heuristic), $z(MILP) = 12$

Job selection

Introduction: Job selection

- It is problem 7.2 where the jobs are no longer compulsory and have a **revenue**
- It becomes a **maximisation** problem: the roles of the bounds swap, the heuristic gives a *lower* bound and the dual an *upper* bound
- The activation link is the same, but the optimality argument changes sign: we will see that the structure of the proof does not change

Problem 7.3: job selection (1/2)

- A company can execute $n \in \mathbb{Z}_{\geq 1}$ jobs and has $k \in \mathbb{Z}_{\geq 1}$ machines
- For each job j, the value $t_j \in \mathbb{Q}_{>0}$ is the processing time in minutes (the same on every machine) and the value $r_j \in \mathbb{Q}_{>0}$ is the revenue in euros if the job is executed
- For each machine m, the value $a_m \in \mathbb{Q}_{>0}$ is the availability in minutes and the value $c_m \in \mathbb{Q}_{>0}$ is the cost in euros if the machine is used

Problem 7.3: job selection (2/2)

- Each machine handles one job at a time
- The company wants to choose which jobs to execute, and on which machines, to **maximise the profit**: revenues of the executed jobs minus costs of the machines used

The problem in words

- **We decide** which jobs to execute, on which machines, and which machines to switch on
- **The objective**: maximum profit
- **The constraints**: every job on at most one machine; no job on a machine switched off; availability respected
- It is problem 7.2 with non-compulsory jobs and a revenue: a **maximisation** problem

Data (input of the model)

Symbol	Type	Meaning
n	$\in \mathbb{Z}_{\geq 1}$	number of jobs, $j \in \{1, 2, \ldots, n\}$
k	$\in \mathbb{Z}_{\geq 1}$	number of machines, $m \in \{1, 2, \ldots, k\}$
t_j	$\in \mathbb{Q}_{>0}$	processing time of job j
r_j	$\in \mathbb{Q}_{>0}$	revenue if job j is executed
a_m	$\in \mathbb{Q}_{>0}$	availability of machine m
c_m	$\in \mathbb{Q}_{>0}$	fixed cost if machine m is used

Decision variables

We introduce the following $n\,k + k$ binary variables:

$$\begin{cases} x_{jm} = 1 \text{ if job } j \text{ is executed by machine } m,\ 0 \text{ otherwise,} \\ y_m = 1 \text{ if machine } m \text{ is used},\ 0 \text{ otherwise,} \end{cases}$$

$$\forall j \in \{1, 2, \ldots, n\},\ \forall m \in \{1, 2, \ldots, k\}.$$

The model

Model 7.3 — job selection

$$\max \quad \sum_{j=1}^{n} \sum_{m=1}^{k} r_j \, x_{jm} - \sum_{m=1}^{k} c_m \, y_m \tag{7.6a}$$

$$\text{subject to} \quad \sum_{m=1}^{k} x_{jm} \leq 1, \qquad \forall j \in \{1, 2, \ldots, n\}, \tag{7.6b}$$

$$\sum_{j=1}^{n} t_j \, x_{jm} - a_m \, y_m \leq 0, \qquad \forall m \in \{1, 2, \ldots, k\}, \tag{7.6c}$$

$$x_{jm} \in \{0, 1\}, \quad \forall j \in \{1, 2, \ldots, n\}, \ \forall m \in \{1, 2, \ldots, k\}, \tag{7.6d}$$

$$y_m \in \{0, 1\}, \qquad \forall m \in \{1, 2, \ldots, k\}. \tag{7.6e}$$

Description of the objective function and constraints

- The objective function maximises the total profit, the difference between the revenue of the executed jobs and the cost of the machines used
- Constraints (1) ensure that each job is assigned to *at most* one machine, so that no job is executed twice (n constraints)
- Constraints (2) link assignments and usage and impose the capacities, exactly as in problem 7.2 (k constraints)
- The last constraints define the variables of the model

The link between the variables (1/2)

It is the link of problem 7.2, with t_j in place of t_{jm}.

- The implication

$$\sum_{j=1}^{n} t_j x_{jm} > 0 \Longrightarrow y_m = 1$$

  says that an unused machine has no jobs

- It is imposed by constraints (2): if $\sum_j t_j x_{jm} > 0$ then $a_m y_m > 0$, hence $y_m = 1$; conversely, if $y_m = 0$ then $\sum_j t_j x_{jm} \leq 0$ and, with $t_j > 0$, all the x_{jm} are zero

The link between the variables (2/2)

- The opposite implication

$$y_m = 1 \Longrightarrow \sum_{j=1}^{n} t_j x_{jm} > 0$$

  says that a used machine has at least one job

- It is not imposed by the constraints and follows from the objective in every optimum: since $c_m > 0$, if $y_m = 1$ with no jobs, setting $y_m = 0$ stays feasible and **increases** the profit by c_m

- In a maximisation problem the direction of the improvement is reversed, the structure of the argument is not

The model in gurobipy

```
m = gp.Model("selection")
x = m.addVars(n, k, vtype=GRB.BINARY, name="x")
y = m.addVars(k, vtype=GRB.BINARY, name="y")
m.setObjective(gp.quicksum(r[j] * x[j, mm]
    for j in range(n) for mm in range(k))
    - gp.quicksum(c[mm] * y[mm] for mm in range(k)),
    GRB.MAXIMIZE)
m.addConstrs((x.sum(j, "*") <= 1 for j in range(n)),
             name="at_most_one")
m.addConstrs((gp.quicksum(t[j] * x[j, mm]
    for j in range(n)) - a[mm] * y[mm] <= 0
    for mm in range(k)), name="link")
```

The instance

$n = 3$ jobs, $k = 3$ machines:

	$m = 1$	$m = 2$	$m = 3$
a_m	105	110	100
c_m	20	30	15

	$j = 1$	$j = 2$	$j = 3$
t_j	25	40	75
r_j	10	15	30

The model for the instance

Model 7.3 on the instance

$$
\begin{aligned}
\max\ & 10x_{11}+10x_{12}+10x_{13}+15x_{21}+15x_{22}+15x_{23}+30x_{31}+30x_{32}+30x_{33}-20y_1-30y_2-15y_3 \\
\text{subject to}\ & x_{11}+x_{12}+x_{13} \le 1 \\
& x_{21}+x_{22}+x_{23} \le 1 \\
& x_{31}+x_{32}+x_{33} \le 1 \\
& 25x_{11}+40x_{21}+75x_{31}-105y_1 \le 0 \\
& 25x_{12}+40x_{22}+75x_{32}-110y_2 \le 0 \\
& 25x_{13}+40x_{23}+75x_{33}-100y_3 \le 0 \\
& x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33} \in \{0,1\} \\
& y_1, y_2, y_3 \in \{0,1\}
\end{aligned}
$$

Constructive heuristic: the primal bound

- In a maximisation problem a feasible solution gives a **lower** bound
- The three heuristics of problem 7.1 adapt with one more rule: a job that fits on no machine is *skipped* (it is not compulsory)
- The best-fit chooses the **fullest** machine among those that are enough (minimum residual availability), to concentrate the jobs on few machines

Run on the instance (1/2)

- **Step 1.** Job 1 ($t_1 = 25$): $ra = (105, 110, 100)$, all enough; the fullest is machine 3: $x[1][3] = 1$, $ra[3] = 75$
- **Step 2.** Job 2 ($t_2 = 40$): $ra = (105, 110, 75)$; the fullest is still machine 3: $x[2][3] = 1$, $ra[3] = 35$
- **Step 3.** Job 3 ($t_3 = 75$): $ra = (105, 110, 35)$; machine 3 is not enough; between 1 and 2 the fullest is machine 1: $x[3][1] = 1$, $ra[1] = 30$

Run on the instance (2/2)

Solution $\bar{x}_{13} = \bar{x}_{23} = \bar{x}_{31} = 1$, $\bar{y} = (1, 0, 1)$, profit

$$10 + 15 + 30 - 20 - 15 = 20, \qquad z(MILP) \geq LB = 20.$$

- Next-fit and first-fit fill machine 1 first (jobs 1 and 2) and put job 3 on machine 2: profit $55 - 50 = 5$

LP relaxation and dual

- Relaxing $x_{jm} \geq 0$ and $y_m \geq 0$, with non-negative duals μ_j for constraints (1) and π_m for constraints (2) ($\leq$ in a maximisation)

Dual of the LP relaxation of model 7.3

$$\min \sum_{j=1}^{n} \mu_j \tag{7.7a}$$

$$\text{subject to} \quad \mu_j + t_j \pi_m \geq r_j, \qquad \forall j \in \{1, 2, \ldots, n\}, \ \forall m \in \{1, 2, \ldots, k\}, \tag{7.7b}$$

$$-a_m \pi_m \geq -c_m, \qquad \forall m \in \{1, 2, \ldots, k\}, \tag{7.7c}$$

$$\mu_j \geq 0, \qquad \forall j \in \{1, 2, \ldots, n\}, \tag{7.7d}$$

$$\pi_m \geq 0, \qquad \forall m \in \{1, 2, \ldots, k\}. \tag{7.7e}$$

Description of the dual

- μ_j is the price of the right to execute job j, π_m that of one unit of capacity of machine m
- The objective adds up the prices of the rights on the jobs
- Constraints (7.7b) are the columns of the x_{jm}: executing job j on machine m consumes the right, at price μ_j, and t_j of capacity, at price π_m; together they must cover the revenue r_j
- Constraints (7.7c) are the columns of the y_m: the capacity a_m made available, priced at π_m, cannot cost more than the fixed cost c_m

The dual of the instance

The instance model

$$
\begin{array}{rrrrrrrrrl}
\min & \mu_1 & +\mu_2 & +\mu_3 & & & & & & \\
\text{subject to} & \mu_1 & & & +25\pi_1 & & & & & \geq 10 \\
& \mu_1 & & & & +25\pi_2 & & & & \geq 10 \\
& \mu_1 & & & & & +25\pi_3 & & & \geq 10 \\
& & \mu_2 & & +40\pi_1 & & & & & \geq 15 \\
& & \mu_2 & & & +40\pi_2 & & & & \geq 15 \\
& & \mu_2 & & & & +40\pi_3 & & & \geq 15 \\
& & & \mu_3 & +75\pi_1 & & & & & \geq 30 \\
& & & \mu_3 & & +75\pi_2 & & & & \geq 30 \\
& & & \mu_3 & & & +75\pi_3 & & & \geq 30 \\
& & & & -105\pi_1 & & & & & \geq -20 \\
& & & & & -110\pi_2 & & & & \geq -30 \\
& & & & & & -100\pi_3 & & & \geq -15 \\
& \mu_1, & \mu_2, & \mu_3 & & & & & & \geq 0 \\
& & & & & \pi_1, & \pi_2, & & \pi_3 & \geq 0
\end{array}
$$

A hand-built dual solution (1/2)

- Constraints (7.7c) bound each π_m from above: $\bar{\pi}_m = c_m/a_m$, that is

$$\bar{\pi}_1 = \tfrac{4}{21}, \qquad \bar{\pi}_2 = \tfrac{3}{11}, \qquad \bar{\pi}_3 = \tfrac{3}{20}$$

- Each μ_j must satisfy $\mu_j \geq r_j - t_j \bar{\pi}_m$ for every m and $\mu_j \geq 0$: the smallest feasible value is the maximum of these

$$\bar{\mu}_1 = \max\left\{\tfrac{110}{21}, \tfrac{35}{11}, \tfrac{25}{4}, 0\right\} = \tfrac{25}{4},$$

$$\bar{\mu}_2 = \max\left\{\tfrac{155}{21}, \tfrac{45}{11}, 9, 0\right\} = 9,$$

$$\bar{\mu}_3 = \max\left\{\tfrac{110}{7}, \tfrac{105}{11}, \tfrac{75}{4}, 0\right\} = \tfrac{75}{4}$$

A hand-built dual solution (2/2)

- The value is $\frac{25}{4} + 9 + \frac{75}{4} = 34$. By weak duality

$$LB = 20 \ \leq \ z(MILP) \ \leq \ UB = 34$$

- The reading: μ_j is "how much job j is worth at most" once the spread fixed cost of the cheapest machine per minute (machine 3, at $\frac{3}{20}$ euros per minute) has been paid for every minute

What the solver says (1/2)

- The optimum of the relaxation without the bounds is $z(LP) = 34$: here too the hand-built solution is optimal for the dual
- With the bounds $x_{jm} \leq 1$ and $y_m \leq 1$ the relaxation drops to $z(LP^+) = 680/21 = 32.38$
- The integer optimum is $z(MILP) = 25$: jobs 1 and 3 on machine 3 ($25 + 75 = 100$, exactly the availability), profit $40 - 15$
- Job 2 is not worth it, since it would require a second machine ($c_1 = 20 > r_2 = 15$)

What the solver says (2/2)

LB (best-fit)	UB (hand dual)	$z(LP)$	$z(LP^+)$	$z(MILP)$
20	34	34	680/21	25

- The heuristic stopped at 20: it had put job 2 on machine 3, leaving out job 3, which then opened machine 1
- Gap of the heuristic with respect to the optimum: 20%

Additional considerations

- $y_m \leq 1$ (k) and $x_{jm} \leq 1$ $(n\,k)$ are valid inequalities; the latter are implied by constraints (1), the former strengthen the relaxation (34 $\rightarrow$ 32.38)
- The disaggregated constraints $x_{jm} \leq y_m$, from the implication $(x_{1m}$ OR $\cdots$ OR $x_{nm}) \Longrightarrow y_m$ already seen in problem 7.2, are valid and implied by the model, but strengthen the relaxation
- If $r_j < \min_m c_m$ and job j is the only one on a machine, executing it never pays: this is what happens to job 2 in the instance

Exercise: all jobs compulsory

- The company has already signed the contracts: all jobs must be executed
- How does the model change?
- How much does the obligation cost on the instance?

The final comparison (1/2)

The three numbers that every problem of the course produces:

	value	what it is
LB	20	heuristic solution
UB	34	certified bound
$z(MILP)$	25	MILP optimum

In a maximisation problem the heuristic gives the **lower** bound and the dual the **upper** bound.

The final comparison (2/2)

- Gap of the heuristic with respect to the optimum:

$$\frac{|LB - z(MILP)|}{|z(MILP)|} = 20.0\%$$

- Certified gap, known *before* solving the MILP: in absolute terms $UB - LB = 14$; as a percentage, against the heuristic solution,

$$\frac{UB - LB}{|LB|} = 70.0\%$$

A worked variant: a job conditional on another (1/2)

- Job 3 requires a semi-finished product made by job 2: job 3 can be executed only if job 2 is executed too (on any machine)
- The implication "3 $\Rightarrow$ 2" reads

$$\sum_{m=1}^{k} x_{3m} \leq \sum_{m=1}^{k} x_{2m}$$

  (one linear constraint)
- If job 3 is executed the left-hand side is 1 and the constraint forces the right-hand side to 1; the constraint does not impose the converse

A worked variant: a job conditional on another (2/2)

- On the instance the optimum becomes 20: the 25 solution (jobs 1 and 3 without job 2) is no longer feasible

	value	what it is
UB	34	hand-built dual certificate
LB	20	heuristic solution
$z(LP)$	34	relaxation without the bounds
$z(LP^+)$	$\frac{680}{21}$	relaxation with the bounds
$z(MILP)$	20	MILP optimum

Summary: Job selection

- Non-compulsory jobs ($\sum_m x_{jm} \leq 1$) with a revenue, machines with a fixed cost
- Same activation link as problem 7.2, with the optimality argument reversed in sign
- In a maximisation problem the heuristic gives LB and the dual gives UB
- Bounds: $LB = 20$ (heuristic), $UB = 34$ (hand dual $= z(LP)$), $z(MILP) = 25$

Parallel jobs

Introduction: Parallel jobs

- The jobs assigned to the same machine start together: the processing time of the machine is the **maximum** of the times of its jobs
- A maximum is not linear, but it is linearised with a continuous variable and n "$\geq$" constraints per machine
- It is the pattern of the **variable defined as a maximum**: we will prove it in three steps

Problem 7.4: parallel jobs (1/2)

- A company needs to execute $n \in \mathbb{Z}_{\geq 1}$ jobs using $k \in \mathbb{Z}_{\geq 1}$ machines
- For each job j and each machine m, the value $t_{jm} \in \mathbb{Q}_{>0}$ is the processing time in minutes of job j on machine m
- Each machine m can execute at most $p_m \in \mathbb{Z}_{\geq 1}$ jobs, and executes them in parallel: all the jobs assigned to the same machine start together

Problem 7.4: parallel jobs (2/2)

- The company wants to assign the jobs minimising the **sum of the processing times of the machines**
- The processing time of a machine is the time of the longest job among those assigned to it

The problem in words

- **We decide** on which machine each job goes
- **The objective**: the sum of the processing times of the machines, each equal to the *maximum* of the times of the assigned jobs
- **The constraints**: every job on one machine; at most p_m jobs on machine m
- The novelty is the objective: a maximum is not linear, but it is linearised

Data (input of the model)

Symbol	Type	Meaning
n	$\in \mathbb{Z}_{\geq 1}$	number of jobs, $j \in \{1, 2, \ldots, n\}$
k	$\in \mathbb{Z}_{\geq 1}$	number of machines, $m \in \{1, 2, \ldots, k\}$
t_{jm}	$\in \mathbb{Q}_{>0}$	processing time of job j on machine m
p_m	$\in \mathbb{Z}_{\geq 1}$	maximum number of (parallel) jobs on machine m

Decision variables

We introduce the following $n\,k$ binary variables and k non-negative variables:

$$\begin{cases} x_{jm} = 1 \text{ if job } j \text{ is executed by machine } m,\ 0 \text{ otherwise,} \\ y_m = \text{processing time of machine } m, \end{cases}$$

$$\forall j \in \{1, 2, \ldots, n\},\ \forall m \in \{1, 2, \ldots, k\}.$$

The model

Model 7.4 — parallel jobs

$$\min \quad \sum_{m=1}^{k} y_m \tag{7.8a}$$

$$\text{subject to} \quad \sum_{m=1}^{k} x_{jm} = 1, \qquad \forall j \in \{1, 2, \ldots, n\}, \tag{7.8b}$$

$$\sum_{j=1}^{n} x_{jm} \leq p_m, \qquad \forall m \in \{1, 2, \ldots, k\}, \tag{7.8c}$$

$$-t_{jm}\, x_{jm} + y_m \geq 0, \qquad \forall j \in \{1, 2, \ldots, n\}, \ \forall m \in \{1, 2, \ldots, k\}, \tag{7.8d}$$

$$x_{jm} \in \{0, 1\}, \quad \forall j \in \{1, 2, \ldots, n\}, \ \forall m \in \{1, 2, \ldots, k\}, \tag{7.8e}$$

$$y_m \geq 0, \qquad \forall m \in \{1, 2, \ldots, k\}. \tag{7.8f}$$

Description of the objective function and constraints

- The objective function minimises the sum of the processing times of the machines
- Constraints (1) ensure that each job is assigned to exactly one machine (n constraints)
- Constraints (2) limit the number of jobs assigned to each machine (k constraints)
- Constraints (3) guarantee that, if the job is assigned to the machine, the processing time of the machine is at least the time required by that job ($n\,k$ constraints)

The link between the variables (1/3)

- For each machine m and each job j:

$$x_{jm} = 1 \implies y_m \geq t_{jm},$$

and by contraposition $y_m < t_{jm} \implies x_{jm} = 0$

- It is imposed by constraints (3), which give $y_m \geq t_{jm} x_{jm}$: if $x_{jm} = 1$, then $y_m \geq t_{jm}$; if $y_m < t_{jm}$, x_{jm} cannot equal 1
- Since it holds for every j, the family (3) imposes, for every machine,

$$y_m \geq \max\{t_{jm} x_{jm} : j \in \{1, 2, \ldots, n\}\}$$

The link between the variables (2/3)

- Conversely, if no job is assigned to machine m, its processing time is zero:

$$\sum_{j=1}^{n} x_{jm} = 0 \implies y_m = 0$$

- It is not imposed by the constraints: with $x_{jm} = 0$ for every j constraints (3) reduce to $y_m \geq 0$
- It follows from the objective in every optimum: y_m has coefficient $1 > 0$; if no job is on the machine and $y_m > 0$, setting $y_m = 0$ keeps the constraints and decreases the objective by $y_m > 0$

The link between the variables (3/3)

Synthesis: in every optimal solution

$$y_m = \max\{t_{jm}x_{jm} : j \in \{1, 2, \ldots, n\}\},$$

i.e. y_m is exactly the processing time of the machine.

- The first point gives $y_m \geq$ maximum in every feasible solution
- If in some solution $y_m >$ maximum, setting y_m equal to the maximum would keep all constraints (3) and decrease the objective by the difference, strictly positive: the solution was not optimal
- With no jobs the maximum is 0, and the second point is recovered

The model in gurobipy

```
m = gp.Model("parallel")
x = m.addVars(n, k, vtype=GRB.BINARY, name="x")
y = m.addVars(k, name="y")                    # continuous, >=
    0
m.setObjective(y.sum(), GRB.MINIMIZE)
m.addConstrs((x.sum(j, "*") == 1 for j in range(n)),
             name="assign")
m.addConstrs((x.sum("*", mm) <= p[mm] for mm in range(
    k)),
             name="cardinality")
m.addConstrs((-t[j][mm] * x[j, mm] + y[mm] >= 0
    for j in range(n) for mm in range(k)), name="
        maximum")
m.optimize()
```

The instance

$n = 3$ jobs, $k = 3$ machines, with the times of problem 7.2:

t_{jm}	$m = 1$	$m = 2$	$m = 3$
$j = 1$	6	5	3
$j = 2$	5	10	2
$j = 3$	20	13	10

	$m = 1$	$m = 2$	$m = 3$
p_m	1	2	2

The model has 9 maximum constraints, one per job–machine pair, the three assignment constraints and the three cardinalities.

The model for the instance

The instance model

$$
\begin{aligned}
\min \quad & y_1+y_2+y_3 \\
\text{subject to} \quad & x_{11}+x_{12}+x_{13} = 1 \\
& x_{21}+x_{22}+x_{23} = 1 \\
& x_{31}+x_{32}+x_{33} = 1 \\
& x_{11}+x_{21}+x_{31} \le 1 \\
& x_{12}+x_{22}+x_{32} \le 2 \\
& x_{13}+x_{23}+x_{33} \le 2 \\
& -6x_{11}+y_1 \ge 0 \\
& -5x_{12}+y_2 \ge 0 \\
& -3x_{13}+y_3 \ge 0 \\
& -5x_{21}+y_1 \ge 0 \\
& -10x_{22}+y_2 \ge 0 \\
& -2x_{23}+y_3 \ge 0 \\
& -20x_{31}+y_1 \ge 0 \\
& -13x_{32}+y_2 \ge 0 \\
& -10x_{33}+y_3 \ge 0 \\
& x_{11},\ x_{12},\ x_{13},\ x_{21},\ x_{22},\ x_{23},\ x_{31},\ x_{32},\ x_{33} \in \{0,1\} \\
& y_1,\ y_2,\ y_3 \ge 0
\end{aligned}
$$

Constructive heuristic: the primal bound

- Next-fit on the cardinalities: machine 1 is filled up to p_1 jobs, then machine 2, and so on
- The time of every machine is the maximum of the assigned times
- It is a constructive heuristic: a single solution, one element at a time, never backtracking

Run on the instance

- **Step 1.** Job 1 on machine 1 (one job $\leq p_1 = 1$):
  $y[1] = \max(0, 6) = 6$
- **Step 2.** Machine 1 is full: job 2 on machine 2,
  $y[2] = \max(0, 10) = 10$
- **Step 3.** Job 3 on machine 2 (two jobs $\leq p_2 = 2$):
  $y[2] = \max(10, 13) = 13$

Solution $\bar{x}_{11} = \bar{x}_{22} = \bar{x}_{32} = 1$, $\bar{y} = (6, 13, 0)$, value 19:

$$z(MILP) \leq UB = 19.$$

LP relaxation and dual

- Relaxing $x_{jm} \geq 0$, with duals μ_j free for (1), $\pi_m \leq 0$ for (2) and $\lambda_{jm} \geq 0$ for (3)

Dual of the LP relaxation of model 7.4

$$\max \quad \sum_{j=1}^{n} \mu_j + \sum_{m=1}^{k} p_m \pi_m \tag{7.9a}$$

$$\text{subject to} \quad \mu_j + \pi_m - t_{jm} \lambda_{jm} \leq 0, \quad \forall j \in \{1, 2, \ldots, n\}, \ \forall m \in \{1, 2, \ldots, k\}, \tag{7.9b}$$

$$\sum_{j=1}^{n} \lambda_{jm} \leq 1, \qquad \forall m \in \{1, 2, \ldots, k\}, \tag{7.9c}$$

$$\mu_j \gtreqless 0, \qquad \forall j \in \{1, 2, \ldots, n\}, \tag{7.9d}$$

$$\pi_m \leq 0, \qquad \forall m \in \{1, 2, \ldots, k\}, \tag{7.9e}$$

$$\lambda_{jm} \geq 0, \quad \forall j \in \{1, 2, \ldots, n\}, \ \forall m \in \{1, 2, \ldots, k\}. \tag{7.9f}$$

Description of the dual

- μ_j is the value of job j, π_m the (non-positive) price of one of the p_m slots of machine m, λ_{jm} the price of the link that pushes the maximum y_m above t_{jm}
- The objective prices the jobs and the available slots
- Constraints (7.9b) are the columns of the x_{jm}: putting job j on machine m satisfies the assignment and takes one slot, but forces the maximum up to t_{jm}; the balance cannot be positive
- Constraints (7.9c) are the columns of the y_m: the maximum costs 1 in the primal objective, and the prices of the links cannot be worth more

The dual of the instance

The instance model

$$
\begin{array}{rcccccccccccccccccccc}
\max & \mu_1 & + \mu_2 & + \mu_3 & + \pi_1 & + 2\pi_2 & + 2\pi_3 & & & & & & & & & & & & & & \\
\text{subject to} & \mu_1 & & & + \pi_1 & & & - 6\lambda_{11} & & & & & & & & & & & \leq & 0 \\
& \mu_1 & & & & + \pi_2 & & & - 5\lambda_{12} & & & & & & & & & & \leq & 0 \\
& \mu_1 & & & & & + \pi_3 & & & - 3\lambda_{13} & & & & & & & & & \leq & 0 \\
& & \mu_2 & & + \pi_1 & & & & & & - 5\lambda_{21} & & & & & & & & \leq & 0 \\
& & \mu_2 & & & + \pi_2 & & & & & & - 10\lambda_{22} & & & & & & & \leq & 0 \\
& & \mu_2 & & & & + \pi_3 & & & & & & - 2\lambda_{23} & & & & & & \leq & 0 \\
& & & \mu_3 & + \pi_1 & & & & & & & & & - 20\lambda_{31} & & & & & \leq & 0 \\
& & & \mu_3 & & + \pi_2 & & & & & & & & & - 13\lambda_{32} & & & \leq & 0 \\
& & & \mu_3 & & & + \pi_3 & & & & & & & & & - 10\lambda_{33} & \leq & 0 \\
& & & & & & & \lambda_{11} & & + \lambda_{21} & & & + \lambda_{31} & & & & \leq & 1 \\
& & & & & & & & \lambda_{12} & & + \lambda_{22} & & & + \lambda_{32} & & \leq & 1 \\
& & & & & & & & & \lambda_{13} & & + \lambda_{23} & & & + \lambda_{33} & \leq & 1 \\
& \mu_1, & \mu_2, & \mu_3 & & & & & & & & & & & & & \gtreqless & 0 \\
& & & & \pi_1, & \pi_2, & \pi_3 & & & & & & & & & & \leq & 0 \\
& & & & & & & \lambda_{11}, & \lambda_{12}, & \lambda_{13}, & \lambda_{21}, & \lambda_{22}, & \lambda_{23}, & \lambda_{31}, & \lambda_{32}, & \lambda_{33} & \geq & 0
\end{array}
$$

A hand-built dual solution

- Constraint (7.9c) is the reduced cost of y_m: the coefficient 1 in the primal objective limits the sum of the λ_{jm}
- Split that unit in equal parts, $\bar{\lambda}_{jm} = 1/n = 1/3$, and set $\bar{\pi}_m = 0$; constraints (7.9b) become $\mu_j \leq t_{jm}/3$ for every m:

$$\bar{\mu}_1 = \frac{\min\{6,5,3\}}{3} = 1, \qquad \bar{\mu}_2 = \frac{\min\{5,10,2\}}{3} = \frac{2}{3},$$

$$\bar{\mu}_3 = \frac{\min\{20,13,10\}}{3} = \frac{10}{3}$$

- Value $1 + \frac{2}{3} + \frac{10}{3} = 5$, hence $LB = 5 \leq z(MILP) \leq UB = 19$

What the solver says (1/2)

- $z(LP) = 520/49 = 10.61$, equal to the optimum of the dual; the relaxation with the bounds coincides, because $x_{jm} \leq 1$ is implied
- The integer optimum is $z(MILP) = 15$: job 1 on machine 2, jobs 2 and 3 on machine 3, $\tilde{y} = (0, 5, 10)$
- The hand bound (5) is far from the relaxation (10.61): the uniform split $\lambda_{jm} = 1/3$ is the first that comes to mind, not the best
- The optimal dual distributes the unit of every machine differently

What the solver says (2/2)

UB (next-fit)	LB (hand dual)	$z(LP)$	$z(MILP)$	heuristic gap
19	5	520/49	15	26.7%

- It is the first problem of the family where the heuristic does *not* find the optimum

Additional considerations

- $x_{jm} \leq 1$ ($n\,k$ valid inequalities) is implied by constraints (1)
- The direction "no job $\Rightarrow y_m = 0$" can be imposed explicitly: with $\bar{t}_m = \max_j t_{jm}$,

$$\bar{t}_m \sum_{j=1}^{n} x_{jm} - y_m \geq 0, \qquad \forall m \in \{1, 2, \ldots, k\}$$

  forces $y_m = 0$ if the machine is empty
- They are *not* valid inequalities, but they are **optimality-preserving constraints**: they can be added because the objective minimises $\sum_m y_m$

Exercise: fixed cost if the machine works

- Switching a machine on costs $g_m = 4$ euros (once) and one minute of processing costs 1 euro
- Minimise the total cost
- Which link is needed and what is the smallest big-M?

The final comparison (1/2)

The three numbers that every problem of the course produces:

	value	what it is
UB	19	heuristic solution
LB	5	certified bound
$z(MILP)$	15	MILP optimum

The final comparison (2/2)

- Gap of the heuristic with respect to the optimum:

$$\frac{|UB - z(MILP)|}{|z(MILP)|} = 26.7\%$$

- Certified gap: in absolute terms $UB - LB = 14$; as a percentage, against the heuristic solution,

$$\frac{UB - LB}{|UB|} = 73.7\%$$

A worked variant: the time of the slowest machine (1/3)

- The machines all work together and the company wants to finish as early as possible: minimise the *maximum* of the processing times of the machines (the makespan), not the sum
- A second level of maximum: a variable $w \geq 0$ with the constraints

$$w \geq y_m, \qquad \forall m \in \{1, 2, \ldots, k\}$$

  and objective min w
- The constraints impose $w \geq \max_m y_m$ and the minimisation forces $w = \max_m y_m$ at the optimum

A worked variant: the time of the slowest machine (2/3)

- **Careful**: now the y_m no longer have a positive coefficient in the objective, so the argument "y_m = maximum of the $t_{jm}x_{jm}$ in every optimum" falls: y_m can inflate up to w at no cost
- It remains true that *there exists* an optimum where y_m is the maximum: just lower it, the constraints stay satisfied and w does not grow
- On the instance the optimal makespan is 10: job 3 requires at least 10 minutes on any machine

A worked variant: the time of the slowest machine (3/3)

	value	what it is
UB	13	heuristic solution
LB	$\frac{260}{59}$	hand-built dual certificate
$z(LP)$	$\frac{260}{59}$	relaxation without the bounds
$z(MILP)$	10	MILP optimum

- In the dual the cost is removed from the y_m and a family $\nu_m \geq 0$ is added for the constraints $w \geq y_m$, with $\sum_m \nu_m \leq 1$
- It pays to price a single job, the longest one, with ν_m proportional to $1/t_{jm}$: the bound is the harmonic mean of its times

Summary: Parallel jobs

- Binary variables x_{jm} and continuous variables y_m defined as a **maximum**
- $y_m \geq t_{jm} x_{jm}$ for every job: the constraints give "$\geq$ maximum", the objective gives the equality
- The direction "empty machine $\Rightarrow y_m = 0$" follows from optimality, not from the constraints
- Bounds: $LB = 5$ (hand dual, far from $z(LP) = 520/49$), $UB = 19$ (next-fit), $z(MILP) = 15$