Mixed problems (1/3)

10.1–10.3

Sapienza Università di Roma

Outline

Why a chapter of mixed problems

Prizes obtainable in two ways

Combinatorial auction

Diet with a count of the foods and a minimum lot

Learning objectives (1/2)

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

- recognise which modelling pieces a problem without a familiar structure combines;
- model selection with alternative modes, covering with packs and splitting with balancing;

Learning objectives (2/2)

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

- explain why the LP relaxation of a splitting problem is weak;
- find a dual bound with a combinatorial argument: parity, number of containers, dominance.

Prerequisites

- From this module, Chapter 5 «Logic and binary variables»: logical constraints between binary variables
- From this module, Chapter 6 «Links between variables»: minimum lot, maximum variable, min-max and max-min
- From this module, Chapter 9 «Production planning»: the bound protocol applied to a family of problems

Why a chapter of mixed problems

Introduction: mixed problems

- The three families of the previous chapters have a recognisable structure: one **assigns**, one **locates**, one **plans**
- The nine problems in this chapter do not have one, and none should be forced on them
- Each puts together different pieces: the modelling work consists precisely in **recognising which**
- This is the chapter where one learns to model without a ready-made scheme

The four groups (1/2)

- **Selection with alternative modes** (problems 10.1 and 10.2)
  - a subset is chosen, but each object has more than one way of being chosen
  - the ways exclude one another
- **Counts with a minimum lot** (problem 10.3)
  - the variables are not binary but integer quantities
  - a quantity may be zero or else above a threshold

The four groups (2/2)

- **Covering with containers** (problems 10.4 and 10.5)
  - a requirement must be covered by buying packs of fixed composition
  - the excess is either paid for or wasted
- **Splitting and balancing** (problems 10.6–10.9)
  - a set must be divided among several containers
  - the quality is measured by how much the containers resemble one another

The new trait: a weak LP relaxation

- Four of these problems share a trait that did not appear in the three families: the **LP relaxation is weak**
- In two cases it is **exactly zero**
- The reason is always the same: a fractional solution can split every object in half and put one half in each container, **levelling everything**
- The useful dual bound must then be looked for elsewhere

Where to look for the dual bound

Three arguments that the dual of the relaxation cannot see:

- **parity**, when a count cannot but be even
- the **number of containers** needed, read off the capacities
- the **dominance of one class** of objects over the others

These are *combinatorial* arguments: they come from integrality, and the dual of the relaxation cannot see them.

The scheme of every problem

For each problem, as in the families:

- description and decision variables
- symbolic model and model on the instance
- a feasible solution with its **primal bound** UB
- the linear relaxation with its dual
- a dual solution with its **dual bound** LB
- the optimal solution $z(MILP)$ and the final comparison

Model class: **BIP** / **MILP**. One Python script per problem.

Summary: why a chapter of mixed problems

- Nine problems with no common structure: modelling means recognising the pieces
- Four groups: alternative modes, minimum lot, covering with containers, splitting and balancing
- In four problems the LP relaxation is weak (in two it is zero)
- The useful bounds come from combinatorial arguments: parity, number of containers, dominance

Prizes obtainable in two ways

Introduction: Prizes with two modes

- A selection problem in which every prize can be obtained in **two alternative ways**: with points only, or with fewer points plus a cash contribution
- We will learn to use **two families of binaries** and the mutual exclusion constraint $x_i + y_i \leq 1$
- We will build a primal bound with a constructive heuristic and a dual bound by hand
- We will see a case in which the dual bound is honest but **very far** from the integer optimum

Problem 10.1: prizes with two modes (1/2)

- A loyalty programme offers $s \in \mathbb{Z}_{\geq 1}$ prizes and a customer has $p \in \mathbb{Q}_{>0}$ points
- Every prize $i \in \{1, 2, \ldots, s\}$ can be obtained in two alternative ways:
  - with points only, spending $a_i \in \mathbb{Q}_{>0}$ of them
  - or spending only $b_i \in \mathbb{Q}_{>0}$ points (with $b_i < a_i$) and adding a cash contribution of $c_i \in \mathbb{Q}_{\geq 0}$ euros
- Every prize can be obtained at most once, and in one mode only

Problem 10.1: prizes with two modes (2/2)

- Every prize carries a preference value $d_i \in \mathbb{Q}_{>0}$
- The customer wants an overall preference of at least $\ell \in \mathbb{Q}_{>0}$
- The customer wants to **minimise the money spent**

The problem in words

- We **decide** which prizes to take and, for each of them, in which of the two modes
- **The objective**: minimum cash contribution
- **The constraints**:
  - the points are not enough for everything
  - the overall preference must reach the threshold
  - the same prize is not taken twice

Data (model input)

Symbol	Type	Meaning
s	$\in \mathbb{Z}_{\geq 1}$	number of prizes, $i \in \{1, 2, \ldots, s\}$
a_i	$\in \mathbb{Q}_{>0}$	points required by the points-only mode
b_i	$\in \mathbb{Q}_{>0}$	points required by the mode with contribution, $b_i < a_i$
c_i	$\in \mathbb{Q}_{\geq 0}$	contribution in euros of the second mode
d_i	$\in \mathbb{Q}_{>0}$	preference value of prize i
p	$\in \mathbb{Q}_{>0}$	points available
ℓ	$\in \mathbb{Q}_{>0}$	minimum preference required

Decision variables

Two families of binaries are needed, one per mode:

$$x_i = \begin{cases} 1 & \text{if prize } i \text{ is taken with points only,} \\ 0 & \text{otherwise,} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if prize } i \text{ is taken with points and contribution,} \\ 0 & \text{otherwise,} \end{cases}$$

for every $i \in \{1, 2, \ldots, s\}$: $2s$ binary variables in total.

Model 10.1

Model 10.1 — prizes with two modes

$$\min \quad \sum_{i=1}^{s} c_i\, y_i \tag{10.1a}$$

$$\text{subject to} \quad x_i + y_i \le 1, \qquad \forall i \in \{1, 2, \ldots, s\}, \tag{10.1b}$$

$$\sum_{i=1}^{s} \left(a_i\, x_i + b_i\, y_i\right) \le p, \tag{10.1c}$$

$$\sum_{i=1}^{s} d_i\, (x_i + y_i) \ge \ell, \tag{10.1d}$$

$$x_i,\ y_i \in \{0, 1\}, \quad \forall i \in \{1, 2, \ldots, s\}. \tag{10.1e}$$

The objective counts euros only: the points-only mode does not appear in it.

Model description

- The **objective function** counts money only: the points-only mode does not appear, because it costs no euros
- The **mutual exclusion** constraints are one per prize (s linear constraints)
- The **points budget** constraint is a single one
- The **preference threshold** constraint is a single one

What mutual exclusion does and does not say (1/2)

The constraint $x_i + y_i \leq 1$ allows *three* configurations: $(0,0)$, $(1,0)$ and $(0,1)$. It only forbids $(1,1)$.

- It does not say "prize i must be taken": the configuration $(0,0)$ is legitimate and means that the prize is given up
- It does not say "if I do not take it with points then I take it with a contribution": the converse

$$x_i = 0 \quad \Longrightarrow \quad y_i = 1$$

is false, and the counterexample is precisely $(0,0)$

What mutual exclusion does and does not say (2/2)

- If one wanted to force *every* prize to be taken, the constraint would have to be written as an equality

$$x_i + y_i = 1$$

- That is a set *partitioning* instead of a set *packing*, and the problem changes
- The quantity $x_i + y_i$ is therefore the **indicator** "prize i has been taken, one way or the other"
- And it is exactly that sum which appears in the preference constraint

The model in gurobipy

```
m = gp.Model("prizes")
x = m.addVars(s, vtype=GRB.BINARY, name="x")
y = m.addVars(s, vtype=GRB.BINARY, name="y")
m.setObjective(gp.quicksum(c[i] * y[i]
                for i in range(s)), GRB.MINIMIZE)
m.addConstrs((x[i] + y[i] <= 1 for i in range(s)),
             name="one_mode")
m.addConstr(gp.quicksum(a[i] * x[i] + b[i] * y[i]
            for i in range(s)) <= p, name="points")
m.addConstr(gp.quicksum(d[i] * (x[i] + y[i])
            for i in range(s)) >= ell, name="preference")
```

The instance

$s = 5$ prizes, $p = 20$ points, $\ell = 16$:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
a_i	8	6	10	5	7
b_i	4	3	6	2	4
c_i	10	8	15	5	9
d_i	5	4	7	3	6

Ten binary variables: five x_i and five y_i.

Model 10.1 on the instance

The instance model

$$
\begin{aligned}
\min \quad & 10y_1+8y_2+15y_3+5y_4+9y_5 \\
\text{subject to} \quad & x_1 + y_1 \le 1 \\
& x_2 + y_2 \le 1 \\
& x_3 + y_3 \le 1 \\
& x_4 + y_4 \le 1 \\
& x_5 + y_5 \le 1 \\
& 8x_1+6x_2+10x_3+5x_4+7x_5+4y_1+3y_2+6y_3+2y_4+4y_5 \le 20 \\
& 5x_1+4x_2+7x_3+3x_4+6x_5+5y_1+4y_2+7y_3+3y_4+6y_5 \ge 16 \\
& x_1, x_2, x_3, x_4, x_5 \in \{0,1\} \\
& y_1, y_2, y_3, y_4, y_5 \in \{0,1\}
\end{aligned}
$$

Constructive heuristic: the primal bound (1/2)

- The prizes are scanned by **decreasing preference**
- Each one is taken with points only if they suffice, otherwise with the contribution if the reduced points suffice, otherwise it is skipped
- One stops as soon as the required preference is reached
- It is a constructive heuristic: a single solution is built, one element at a time, never backtracking

Constructive heuristic: the primal bound (2/2)

On the instance the order by preference is $3, 5, 1, 2, 4$.

- Prize 3 (preference 7): the points alone are enough ($10 \leq 20$), it is taken; 10 points are left, preference 7
- Prize 5 (preference 6): the points alone are enough ($7 \leq 10$), it is taken; 3 points are left, preference 13
- Prize 1 (preference 5): the points are not enough for mode a ($8 > 3$) nor for mode b ($4 > 3$): it is skipped
- Prize 2 (preference 4): not enough for mode a ($6 > 3$), enough for mode b ($3 \leq 3$): it is taken for 8 euros; the preference reaches $17 \geq 16$ and we stop

$$z(MILP) \leq UB = 8$$

Dual of the LP relaxation of model 10.1

We associate $\sigma_i \geq 0$ with mutual exclusion, $\pi \geq 0$ with the points budget and $\rho \geq 0$ with the preference threshold.

Dual of the LP relaxation of model 10.1

$$\max \quad -\sum_{i=1}^{s} \sigma_i - p\,\pi + \ell\,\rho \tag{10.2a}$$

$$\text{subject to} \quad -\sigma_i - a_i\,\pi + d_i\,\rho \leq 0, \quad \forall i \in \{1,2,\ldots,s\}, \tag{10.2b}$$

$$-\sigma_i - b_i\,\pi + d_i\,\rho \leq c_i, \quad \forall i \in \{1,2,\ldots,s\}, \tag{10.2c}$$

$$\sigma_i \geq 0, \quad \forall i \in \{1,2,\ldots,s\}, \tag{10.2d}$$

$$\pi \geq 0, \tag{10.2e}$$

$$\rho \geq 0. \tag{10.2f}$$

Dual on the instance

The instance model

$$
\begin{aligned}
\max\ & -\sigma_1 - \sigma_2 - \sigma_3 - \sigma_4 - \sigma_5 - 20\pi + 16\rho \\
\text{subject to}\ & -\sigma_1 - 8\pi + 5\rho \le 0 \\
& -\sigma_2 - 6\pi + 4\rho \le 0 \\
& -\sigma_3 - 10\pi + 7\rho \le 0 \\
& -\sigma_4 - 5\pi + 3\rho \le 0 \\
& -\sigma_5 - 7\pi + 6\rho \le 0 \\
& -\sigma_1 - 4\pi + 5\rho \le 10 \\
& -\sigma_2 - 3\pi + 4\rho \le 8 \\
& -\sigma_3 - 6\pi + 7\rho \le 15 \\
& -\sigma_4 - 2\pi + 3\rho \le 5 \\
& -\sigma_5 - 4\pi + 6\rho \le 9 \\
& \sigma_1,\ \sigma_2,\ \sigma_3,\ \sigma_4,\ \sigma_5 \ge 0 \\
& \pi \ge 0 \\
& \rho \ge 0
\end{aligned}
$$

Dual description

- π is the **price of one point**, ρ the **value of one unit of preference**, σ_i the price of the mutual exclusion of prize i
- The **objective** collects the threshold ℓ priced at ρ and pays the budget p priced at π together with the multipliers σ_i
- **Columns of the x_i**: taking the prize with points only yields $d_i\,\rho$, consumes $a_i\,\pi$ and σ_i, and the balance cannot exceed the money cost of that option, which is zero
- **Columns of the y_i**: the second option consumes only b_i points and the balance may reach c_i

A dual solution by hand (1/2)

- Here the recipe has *two* free parameters: the price π of a point and the price ρ of one unit of preference
- Once these are fixed, the multipliers of the mutual exclusion follow by taking the smallest feasible value

$$\bar{\sigma}_i = \max\big(0,\ d_i\,\rho - a_i\,\pi\big)$$

- Only the constraints on the second mode remain to be checked
- The function

$$V(\pi,\rho) = -\sum_{i=1}^{s} \bar{\sigma}_i(\pi,\rho) - p\,\pi + \ell\,\rho$$

is concave and piecewise linear: it can be explored on a grid

A dual solution by hand (2/2)

On the instance the maximum is at

$$\bar{\pi} = 2, \qquad \bar{\rho} = 3, \qquad \bar{\sigma} = (0,\ 0,\ 1,\ 0,\ 4)$$

with value

$$LB = -(0 + 0 + 1 + 0 + 4) - 20 \cdot 2 + 16 \cdot 3 = -5 - 40 + 48 = 3$$

- This solution is *optimal* for the relaxation without the bounds: indeed

$$z(LP) = z(LP^{+}) = 3$$

An honest bound can be very far away

Observation 1: An honest bound can be very far away

Here $LB = 3$ and $z(MILP) = 5$: the gap between the dual bound and the integer optimum is 40%, and the certified gap between heuristic and dual is 100%.

Nothing is wrong: the LP relaxation may take "half a prize" at half preference, and that freedom is worth a lot. It is the most extreme case in the course, and it is a reminder that a valid bound is not automatically a useful bound.

Optimal solution

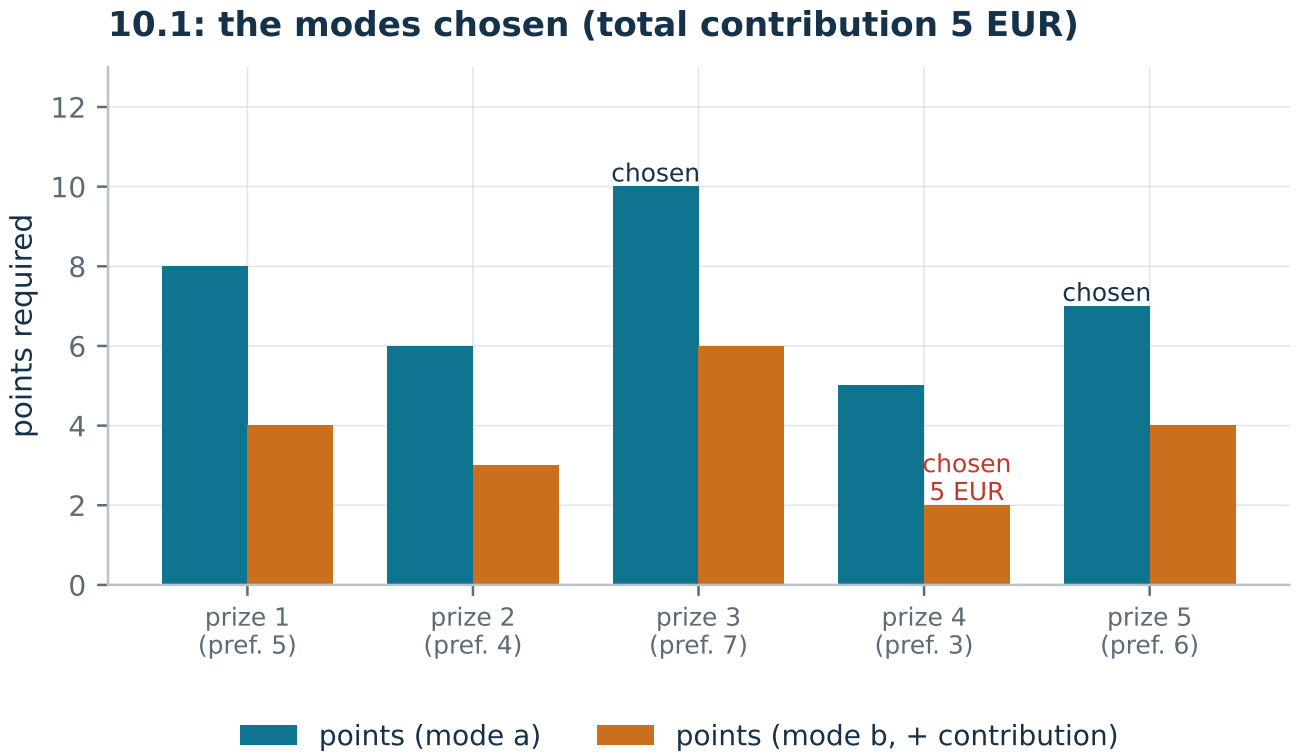
	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
points only x_i	0	0	1	0	1
with contribution y_i	0	0	0	1	0

Prizes 3 and 5 with points only ($10+7=17$) and prize 4 with the contribution (2 points and 5 euros): 19 points out of 20 are used, the preference is $7+6+3=16$ and

$$z(MILP)=5, \qquad 3=LB=z(LP) \leq z(MILP)=5 \leq UB=8$$

The heuristic goes wrong because it only looks at the preference: it is left with 3 points and has to buy prize 2 for 8 euros, whereas the optimum saves the points for prize 4, which costs 5.

The optimum in a picture



For every prize, the points required by the two modes; the labels mark which mode is chosen at the optimum and the contribution paid.

Additional remarks

- The valid inequalities $x_i \leq 1$ and $y_i \leq 1$ are implied by the mutual exclusion constraints: indeed $z(LP) = z(LP^+)$
- If for some prize we had $c_i = 0$, the second mode would dominate the first one (fewer points, same cost) and the variable x_i could be removed: it is a check on the data that is worth doing
- The model extends effortlessly to k modes: k families of binaries and the constraint $\sum_{m=1}^{k} x_i^{(m)} \leq 1$. The structure stays a set packing by rows

Modelling question: alternative prizes

Prizes 3 and 5 come from the same supplier and are alternatives: at most one of them may be taken, in either mode.

How does the model change? What is the new optimum?

The final comparison

The three numbers that every problem of the course produces:

	value	what it is
UB	8	heuristic solution
LB	3	certified bound
$z(MILP)$	5	MILP optimum

- Gap of the heuristic with respect to the optimum: $\dfrac{|UB - z(MILP)|}{|z(MILP)|} = 60.0\%$
- Certified gap, known *before* solving the MILP: $UB - LB = 5$, that is $\dfrac{UB - LB}{|UB|} = 62.5\%$

Worked variant: at least four prizes (1/3)

Besides the preference threshold, the customer wants at least four different prizes. One adds the single linear constraint

$$\sum_{i=1}^{s}(x_i + y_i) \geq 4$$

- Again written on the indicators $x_i + y_i$
- On the instance the optimum rises to 13: with only 20 points, four prizes force a repeated use of the mode that is cheap in points, that is the one that costs money

Worked variant: at least four prizes (2/3)

- The preference constraint and the counting constraint are **not** equivalent, however similar they look: the first one weights the prizes by d_i, the second one counts them
- In the dual the new constraint adds a $\kappa \geq 0$ with right-hand side 4: it enters the objective and appears in *both* columns of every prize, because a prize counts however it was taken
- Searching κ too on the grid of quarters gives $\pi = \rho = 3/2$, $\kappa = 9/2$, that is $LB = 6$
- The heuristic enumerates the $3^5 = 243$ possible states and finds 13, which is the optimum

Worked variant: at least four prizes (3/3)

	value	what it is
UB	13	heuristic solution
LB	6	dual certificate built by hand
$z(LP)$	$\frac{25}{2}$	relaxation without the bounds
$z(LP^+)$	$\frac{25}{2}$	relaxation with the bounds
$z(MILP)$	13	MILP optimum

With five objects enumeration is the most honest rule: the sandwich closes from above, not from below.

Summary: Prizes with two modes

- **Two families of binaries**, one per mode; the sum $x_i + y_i$ is the indicator "prize taken"
- **Mutual exclusion** $x_i + y_i \leq 1$: it only forbids $(1, 1)$, it does not force the prize to be taken
- **The bounds**: heuristic $UB = 8$, dual by hand $LB = 3 = z(LP) = z(LP^+)$, optimum $z(MILP) = 5$
- Heuristic gap 60%, certified gap 62.5%: a valid bound is not automatically a useful bound
- **Variant** with at least four prizes: $z(MILP) = 13$ and $LB = 6$

Combinatorial auction

Introduction: Combinatorial auction

- An auctioneer has items to sell and bids that ask for **groups** of items: it is all or nothing
- It is pure **set packing**: one constraint per item, one variable per bid
- We will learn why conflicts are better written **by rows** and not by pairs
- We will build a primal bound with a greedy heuristic and a dual bound from the prices of the items

Problem 10.2: combinatorial auction (1/2)

- An auctioneer has a set $S = \{1, 2, \ldots, n\}$ of $n \in \mathbb{Z}_{\geq 1}$ items to sell
- He has received $r \in \mathbb{Z}_{\geq 1}$ bids
- For every bid $j \in \{1, 2, \ldots, r\}$:
  - $B_j \subseteq S$ is the subset of items requested
  - $p_j \in \mathbb{Q}_{>0}$ is the profit in euros if the bid is accepted

Problem 10.2: combinatorial auction (2/2)

- Each item can be sold at most once
- A bid can be accepted only if *all* its items are available
- The auctioneer wants to select a set of bids of **maximum total profit**

The problem in words

- We **decide** which bids to accept
- **The objective**: maximum profit
- **The constraint**: two accepted bids cannot ask for the same item
- A bid is *all or nothing*: one does not accept part of it

Data (model input)

Symbol	Type	Meaning
n	$\in \mathbb{Z}_{\geq 1}$	number of items, $i \in S = \{1, 2, \ldots, n\}$
r	$\in \mathbb{Z}_{\geq 1}$	number of bids, $j \in \{1, 2, \ldots, r\}$
B_j	$\subseteq S$	items requested by bid j
p_j	$\in \mathbb{Q}_{>0}$	profit of bid j

Decision variables

A single family of r binary variables:

$$x_j = \begin{cases} 1 & \text{if bid } j \text{ is accepted,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall j \in \{1, 2, \ldots, r\}$$

No variables on the items: the items appear only in the constraints, one per row.

Model 10.2

Model 10.2 — combinatorial auction

$$\max \quad \sum_{j=1}^{r} p_j x_j \tag{10.3a}$$

$$\text{subject to} \quad \sum_{j:\, i \in B_j} x_j \leq 1, \qquad \forall i \in S, \tag{10.3b}$$

$$x_j \in \{0,1\}, \quad \forall j \in \{1,2,\ldots,r\}. \tag{10.3c}$$

Model description

- It is a pure **set packing**: one constraint per item (n linear constraints) and one variable per bid
- The row constraint says that item i is sold **at most once**
- The sum runs only over the bids that ask for that item
- No constraint forces a sale: the constraints are $\leq$, not $=$

By rows, not by pairs

- An alternative, and worse, way is to list all the *pairs* of conflicting bids: $x_j + x_{j'} \leq 1$ for every $j \neq j'$ with $B_j \cap B_{j'} \neq \emptyset$
- On the instance that would be nine constraints instead of four; on real instances the pairs grow like r^2, while the constraints per item stay n
- The row form is also **tighter**: if three bids ask for item i, the row says $x_1 + x_2 + x_3 \leq 1$, whereas the three pairwise inequalities allow $x_1 = x_2 = x_3 = 1/2$
- It is the comparison between aggregated and disaggregated form already seen, but here the direction of the advantage is reversed

The model in gurobipy

```
m = gp.Model("auction")
x = m.addVars(r, vtype=GRB.BINARY, name="x")
m.setObjective(gp.quicksum(p[j] * x[j]
                for j in range(r)), GRB.MAXIMIZE)
m.addConstrs((gp.quicksum(x[j] for j in range(r)
                          if i in B[j]) <= 1
              for i in range(n)), name="item")
```

One row per item, built by scanning the bids that ask for it.

The instance

$n = 4$ items, $r = 6$ bids:

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
B_j	$\{1\}$	$\{2\}$	$\{3, 4\}$	$\{1, 3\}$	$\{2, 4\}$	$\{1, 3, 4\}$
p_j	6	3	12	12	10	16

Six binary variables and four row constraints.

Model 10.2 on the instance

The instance model

$$
\begin{aligned}
\max\ & 6x_1 + 3x_2 + 12x_3 + 12x_4 + 10x_5 + 16x_6 \\
\text{subject to}\ & x_1 + x_4 + x_6 \leq 1 \\
& x_2 + x_5 \leq 1 \\
& x_3 + x_4 + x_6 \leq 1 \\
& x_3 + x_5 + x_6 \leq 1 \\
& x_1, x_2, x_3, x_4, x_5, x_6 \in \{0, 1\}
\end{aligned}
$$

Constructive heuristic: the primal bound (1/2)

- The problem is a **maximisation**, so the heuristic gives the *primal* bound, which sits **below** the optimum
- The bids are sorted by decreasing profit *per item*, that is by $p_j/|B_j|$
- Those whose items are still free are accepted
- It is a constructive heuristic: a single solution, one element at a time, never backtracking. Running time $O(r \log r + r\, n)$

Constructive heuristic: the primal bound (2/2)

The ratios are 6, 3, 6, 6, 5 and $16/3 \approx 5.33$, so the order is $1, 3, 4, 6, 5, 2$:

- bid 1 $\{1\}$: accepted (profit 6)
- bid 3 $\{3, 4\}$: accepted (profit 12)
- bid 4 $\{1, 3\}$: rejected, items 1 and 3 are already sold
- bid 6 $\{1, 3, 4\}$: rejected
- bid 5 $\{2, 4\}$: rejected, item 4 is already sold
- bid 2 $\{2\}$: accepted (profit 3)

Bids $1, 2, 3$ are accepted:

$$z(MILP) \geq LB = 21$$

Dual of the LP relaxation of model 10.2

We associate a non-negative dual variable λ_i with each row constraint.

Dual of the LP relaxation of model 10.2

$$\min \quad \sum_{i \in S} \lambda_i \tag{10.4a}$$

$$\text{subject to} \quad \sum_{i \in B_j} \lambda_i \geq p_j, \quad \forall j \in \{1, 2, \ldots, r\}, \tag{10.4b}$$

$$\lambda_i \geq 0, \qquad \forall i \in S. \tag{10.4c}$$

Dual on the instance

The instance model

$$
\begin{aligned}
\min \quad & \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 & \\
\text{subject to} \quad & \lambda_1 & \geq 6 \\
& \lambda_2 & \geq 3 \\
& \lambda_3 + \lambda_4 & \geq 12 \\
& \lambda_1 + \lambda_3 & \geq 12 \\
& \lambda_2 + \lambda_4 & \geq 10 \\
& \lambda_1 + \lambda_3 + \lambda_4 & \geq 16 \\
& \lambda_1, \ \lambda_2, \ \lambda_3, \ \lambda_4 & \geq 0
\end{aligned}
$$

Dual description

- The variable λ_i is the **price** the auctioneer puts on item i
- The **objective** is the total value of the lots at those prices, and we want it as small as possible
- The **constraints** are the columns of the x_j, one per bid: the items bid j asks for, priced at λ_i, must be worth at least what the bid pays, namely p_j
- In short: no bid may turn out to be a bargain

A dual solution by hand (1/2)

Spread the profit of every bid over its items and take, for every item, the maximum over the bids that ask for it:

$$\bar{\lambda}_i = \max_{j:\, i \in B_j} \frac{p_j}{|B_j|}$$

Feasibility is immediate: for every bid j,

$$\sum_{i \in B_j} \bar{\lambda}_i \;\geq\; \sum_{i \in B_j} \frac{p_j}{|B_j|} = |B_j| \cdot \frac{p_j}{|B_j|} = p_j$$

A dual solution by hand (2/2)

On the instance:

$$\bar{\lambda}_1 = \max\left(\frac{6}{1}, \frac{12}{2}, \frac{16}{3}\right) = 6, \qquad \bar{\lambda}_2 = \max\left(\frac{3}{1}, \frac{10}{2}\right) = 5$$

$$\bar{\lambda}_3 = \max\left(\frac{12}{2}, \frac{12}{2}, \frac{16}{3}\right) = 6, \qquad \bar{\lambda}_4 = \max\left(\frac{12}{2}, \frac{10}{2}, \frac{16}{3}\right) = 6$$

and therefore

$$z(MILP) \leq UB = 6 + 5 + 6 + 6 = 23$$

A crude recipe costs a lot

Observation 2: A crude recipe costs a lot

The "obvious" recipe $\bar{\lambda}_i = \max_{j:i\in B_j} p_j$ *(without dividing by* $|B_j|$*) is feasible as well, but gives* $16+10+16+16=58$*: more than twice as much.*

Dividing by the cardinality of the requested set is what makes the bound useful. It is worth trying several recipes and keeping the best: they are all valid, not all informative.

Optimal solution

At the optimum bids 4 ($\{1,3\}$, profit 12) and 5 ($\{2,4\}$, profit 10) are accepted: all four items are sold and

$$z(MILP) = 22$$

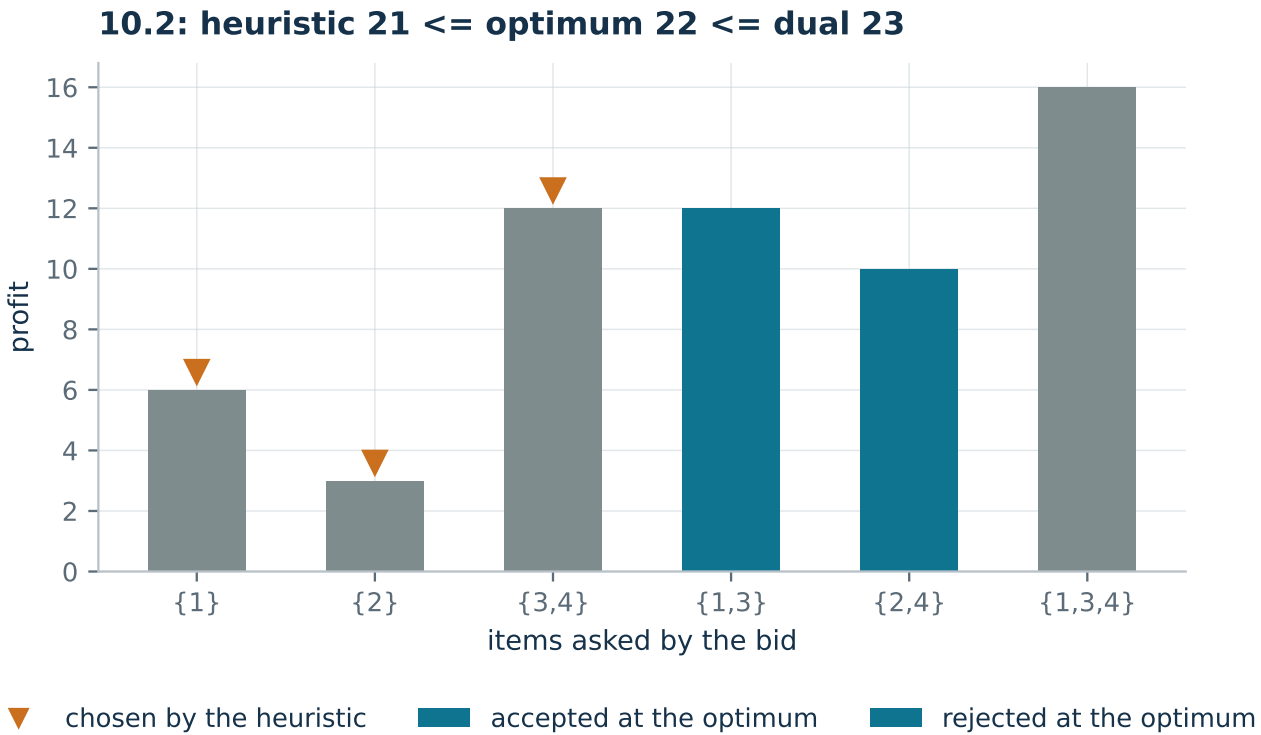
The chain of bounds is

$$21 = LB \leq z(MILP) = 22 = z(LP^{+}) = z(LP) \leq UB = 23$$

with a heuristic gap of 4.5%.

- The auctioneer sells everything, but not because a constraint forces it: the constraints are $\leq$, not $=$
- With other bids the optimal solution could leave items on the shelf

The bids in a picture



The six bids with their profit: in teal those accepted at the optimum, in grey those rejected; the triangle marks those chosen by the heuristic.

Additional remarks (1/2)

- The valid inequalities $x_j \leq 1$ are implied by the row constraints whenever $B_j \neq \emptyset$: indeed $z(LP) = z(LP^+) = 22$
- On this instance one also has $z(LP) = z(MILP)$: the relaxation lands on an **integer vertex**
- It is a lucky case, not a property of set packing. The minimal counterexample is the *triangle*: three items and three bids asking for two of them each, all with profit 1. There $z(LP) = 3/2$ (with $x = 1/2$ on all three) against $z(MILP) = 1$

Additional remarks (2/2)

- Set packing is NP-hard in general, but it becomes easy when the incidence matrix is *perfect* or *balanced*
- The triangle is precisely the smallest graph that is not perfect in this sense
- And it is the reason why *clique inequalities* are the classic cuts for this family

Modelling question: bids from the same bidder

Bids 4 and 5 come from the same bidder, who may win at most one of them.

How does the model change? What is the new optimum?

SAPIENZA UNIVERSITÀ DI ROMA

The final comparison

The three numbers that every problem of the course produces:

	value	what it is
LB	21	heuristic solution
UB	23	certified bound
$z(MILP)$	22	MILP optimum

- Gap of the heuristic with respect to the optimum: $\dfrac{|LB - z(MILP)|}{|z(MILP)|} = 4.5\%$
- Certified gap, known *before* solving the MILP: $UB - LB = 2$, that is $\dfrac{UB - LB}{|LB|} = 9.5\%$

Worked variant: limited deliveries (1/3)

In this round the auctioneer can deliver at most two items in total. One adds the single linear constraint

$$\sum_{j=1}^{r} |B_j| \, x_j \leq 2$$

- The coefficient of every bid is the number of items it asks for: it is a **knapsack** constraint laid on top of the set packing
- On the instance the optimum drops to 12: a single two-item bid is accepted, the more profitable between bid 3 and bid 4 (both worth 12)

Worked variant: limited deliveries (2/3)

- The cap on the items delivered adds to the dual a $\mu \geq 0$ with right-hand side 2, and in the column of every bid it appears multiplied by how many items that bid asks for
- With μ fixed, every bid spreads over its items net of μ and λ_i is the largest among the bids asking for item i
- Trying μ among the per-item profits one reaches $\mu = 6$, $\lambda = 0$ and $UB = 12$
- The heuristic accepts bid 3 alone and is worth 12 as well: the sandwich closes, and the optimum is known without solving the MILP

Worked variant: limited deliveries (3/3)

	value	what it is
UB	12	dual certificate built by hand
LB	12	heuristic solution
$z(LP)$	12	relaxation without the bounds
$z(LP^{+})$	12	relaxation with the bounds
$z(MILP)$	12	MILP optimum

Here the two sides of the sandwich coincide: the certified gap is zero.

Summary: Combinatorial auction

- Pure **set packing**: one binary per bid, one ≤ 1 constraint per item
- Conflicts are written **by rows**, not by pairs: fewer constraints and a tighter relaxation
- **The bounds**: heuristic $LB = 21$, dual by hand $UB = 23$, optimum $z(MILP) = 22 = z(LP) = z(LP^+)$
- Heuristic gap 4.5%, certified gap 9.5%; the crude dual recipe would give 58
- **Variant** with at most two items delivered: all the values drop to 12

Diet with a count of the foods and a minimum lot

Introduction: Diet with variety

- We choose **which** foods go into the diet and **how many** kilos to buy
- Every chosen food must be bought at least in a **minimum lot**: this is what makes the count honest
- We will see the **semicontinuous variable**: either it is zero, or it lies between c_i and d_i
- We will build a primal bound with a constructive heuristic and a dual bound by hand: it is the tightest sandwich of the chapter

Problem 10.3: diet with variety (1/2)

- A nutritionist must compose a monthly diet choosing among $s \in \mathbb{Z}_{\geq 1}$ foods and controlling $r \in \mathbb{Z}_{\geq 1}$ nutrients
- For every food $i \in \{1, 2, \ldots, s\}$, the value $w_i \in \mathbb{Q}_{>0}$ is the cost of one kilo
- For every food i and every nutrient $j \in \{1, 2, \ldots, r\}$, the value $g_{ij} \in \mathbb{Q}_{\geq 0}$ is the amount of nutrient j contained in one kilo of food i

Problem 10.3: diet with variety (2/2)

- For every nutrient j, the monthly intake must lie between $a_j \in \mathbb{Q}_{>0}$ and $b_j \in \mathbb{Q}_{>0}$
- If a food enters the diet, at least $c_i \in \mathbb{Q}_{>0}$ kilos of it must be consumed and at most $d_i \in \mathbb{Q}_{>0}$
- The diet must include at least $t \in \mathbb{Z}_{\geq 1}$ different foods
- The goal is the diet of **minimum cost**

The problem in words

- We **decide** which foods to use and in what quantity
- **The objective**: minimum cost
- **The constraints**:
  - every nutrient within its window
  - every chosen food in a quantity between the minimum lot and the cap
  - at least t different foods

Data (model input)

Symbol	Type	Meaning
s	$\in \mathbb{Z}_{\geq 1}$	number of foods, $i \in \{1, 2, \ldots, s\}$
r	$\in \mathbb{Z}_{\geq 1}$	number of nutrients, $j \in \{1, 2, \ldots, r\}$
w_i	$\in \mathbb{Q}_{>0}$	cost of one kilo of food i
g_{ij}	$\in \mathbb{Q}_{\geq 0}$	grams of nutrient j per kilo of food i
a_j, b_j	$\in \mathbb{Q}_{>0}$	monthly minimum and maximum of nutrient j
c_i, d_i	$\in \mathbb{Q}_{>0}$	minimum and maximum quantity of food i, if chosen
t	$\in \mathbb{Z}_{\geq 1}$	minimum number of different foods

Decision variables

s non-negative continuous variables:

x_i = kilos of food i consumed in the month, $\quad \forall i \in \{1, 2, \ldots, s\}$

s binary variables:

$$y_i = \begin{cases} 1 & \text{if food } i \text{ enters the diet} \\ 0 & \text{otherwise} \end{cases}$$

Model 10.3

Model 10.3 — diet with variety

$$\min \quad \sum_{i=1}^{s} w_i\, x_i \tag{10.5a}$$

$$\text{subject to} \quad \sum_{i=1}^{s} g_{ij}\, x_i \geq a_j, \qquad \forall j \in \{1, 2, \ldots, r\}, \tag{10.5b}$$

$$\sum_{i=1}^{s} g_{ij}\, x_i \leq b_j, \qquad \forall j \in \{1, 2, \ldots, r\}, \tag{10.5c}$$

$$x_i - c_i\, y_i \geq 0, \qquad \forall i \in \{1, 2, \ldots, s\}, \tag{10.5d}$$

$$x_i - d_i\, y_i \leq 0, \qquad \forall i \in \{1, 2, \ldots, s\}, \tag{10.5e}$$

$$\sum_{i=1}^{s} y_i \geq t, \tag{10.5f}$$

$$x_i \geq 0, \qquad \forall i \in \{1, 2, \ldots, s\}, \tag{10.5g}$$

$$y_i \in \{0, 1\}, \qquad \forall i \in \{1, 2, \ldots, s\}. \tag{10.5h}$$

Model description

- The **objective function** is the total cost
- The **minimum requirement** constraints, one per nutrient, and the **cap** constraints, again one per nutrient, close every nutrient inside its window
- The **minimum lot** and **activation** constraints, one per food each, make x_j a **semicontinuous** variable: either it is zero, or it lies between c_j and d_j
- The **variety** constraint, a single one, requires at least t different foods to be bought

Why the count needs the minimum lot

- The variety constraint counts the indicators y_i, not the quantities
- With the activation constraint alone one could have $y_i = 1$ with $x_i = 0$: foods switched on without a single gram consumed, and variety satisfied by empty indicators
- The minimum lot closes the loop: with $y_i = 1$ one gets $x_i \geq c_i > 0$, so every switched-on indicator is a food really consumed
- On the instance, setting $c_i = 0$ the optimum drops from $48/5$ to $46/5$ and four foods are switched on, two with zero quantity: the count no longer says anything

The model in gurobipy (1/2)

```
m = gp.Model("diet")
x = m.addVars(s, name="x")
y = m.addVars(s, vtype=GRB.BINARY, name="y")
m.setObjective(gp.quicksum(w[i] * x[i]
                for i in range(s)), GRB.MINIMIZE)
m.addConstrs((gp.quicksum(g[i][j] * x[i]
              for i in range(s)) >= a[j]
              for j in range(r)), name="minimum")
```

The model in gurobipy (2/2)

```
m.addConstrs((gp.quicksum(g[i][j] * x[i]
              for i in range(s)) <= b[j]
              for j in range(r)), name="maximum")
m.addConstrs((x[i] - c[i] * y[i] >= 0
              for i in range(s)), name="minimum_lot")
m.addConstrs((x[i] - d[i] * y[i] <= 0
              for i in range(s)), name="activate")
m.addConstr(gp.quicksum(y[i] for i in range(s)) >= t,
            name="variety")
```

The instance

$s = 4$ foods, $r = 2$ nutrients, $c_i = 1$, $d_i = 8$, $t = 3$:

	milk	rice	bread	potatoes
w_i (euros/kg)	2	3	1	4
iron (g/kg)	10	20	5	25
calcium (g/kg)	5	10	15	5

	iron	calcium
a_j	60	40
b_j	200	150

Four continuous variables x_i and four binaries y_i.

Model 10.3 on the instance

The instance model

$$
\begin{array}{rrrrrrrrrrrr}
\min & 2x_1 & +3x_2 & +x_3 & +4x_4 & & & & & & & \\
\text{subject to} & 10x_1 & +20x_2 & +5x_3 & +25x_4 & & & & & & \geq & 60 \\
& 5x_1 & +10x_2 & +15x_3 & +5x_4 & & & & & & \geq & 40 \\
& 10x_1 & +20x_2 & +5x_3 & +25x_4 & & & & & & \leq & 200 \\
& 5x_1 & +10x_2 & +15x_3 & +5x_4 & & & & & & \leq & 150 \\
& x_1 & & & & -y_1 & & & & & \geq & 0 \\
& & x_2 & & & & -y_2 & & & & \geq & 0 \\
& & & x_3 & & & & -y_3 & & & \geq & 0 \\
& & & & x_4 & & & & -y_4 & & \geq & 0 \\
& x_1 & & & & -8y_1 & & & & & \leq & 0 \\
& & x_2 & & & & -8y_2 & & & & \leq & 0 \\
& & & x_3 & & & & -8y_3 & & & \leq & 0 \\
& & & & x_4 & & & & -8y_4 & & \leq & 0 \\
& & & & & y_1 & +y_2 & +y_3 & +y_4 & & \geq & 3 \\
& x_1, & x_2, & x_3, & x_4 & & & & & & \geq & 0 \\
& & & & & y_1, & y_2, & y_3, & y_4 & & \in & \{0,1\}
\end{array}
$$

Constructive heuristic: the primal bound (1/2)

- The t cheapest foods are switched on at their **minimum lot**
- Then the residual requirement of each nutrient is covered by adding the already active food with the lowest cost per gram

It is a constructive heuristic: a single solution, one element at a time, never backtracking.

On the instance the three cheapest foods are bread (1), milk (2) and rice (3): switched on at their minimum lot, that is 1 kg each.

iron: $5 + 10 + 20 = 35$ g (60 needed), calcium: $15 + 5 + 10 = 30$ g (40 needed)

Constructive heuristic: the primal bound (2/2)

- **Iron:** 25 grams are missing. Among the active foods the lowest cost per gram of iron is that of rice, $3/20 = 0.15$; $25/20 = 1.25$ extra kilos are needed and the rice reaches 2.25 kg
- **Calcium:** after that addition one has $30 + 12.5 = 42.5$ grams, already above the minimum of 40: nothing else is needed

The final diet is milk 1 kg, rice 2.25 kg, bread 1 kg, at a cost of

$$z(MILP) \leq UB = \frac{39}{4} = 9.75$$

Dual of the LP relaxation of model 10.3

$\alpha_j \geq 0$ with the minima, $\beta_j \geq 0$ with the maxima, $\lambda_i \geq 0$ with the minimum lots, $\mu_i \geq 0$ with the caps, $\tau \geq 0$ with the variety.

Dual of the LP relaxation of model 10.3

$$\max \quad \sum_{j=1}^{r} a_j \alpha_j - \sum_{j=1}^{r} b_j \beta_j + t\tau \tag{10.6a}$$

$$\text{subject to} \quad \sum_{j=1}^{r} g_{ij} (\alpha_j - \beta_j) + \lambda_i - \mu_i \leq w_i, \quad \forall i \in \{1, 2, \ldots, s\}, \tag{10.6b}$$

$$-c_i \lambda_i + d_i \mu_i + \tau \leq 0, \quad \forall i \in \{1, 2, \ldots, s\}, \tag{10.6c}$$

$$\alpha_j \geq 0, \quad \forall j \in \{1, 2, \ldots, r\}, \tag{10.6d}$$

$$\beta_j \geq 0, \quad \forall j \in \{1, 2, \ldots, r\}, \tag{10.6e}$$

$$\lambda_i \geq 0, \quad \forall i \in \{1, 2, \ldots, s\}, \tag{10.6f}$$

$$\mu_i \geq 0, \quad \forall i \in \{1, 2, \ldots, s\}, \tag{10.6g}$$

$$\tau \geq 0. \tag{10.6h}$$

Dual on the instance

The instance model

$$
\begin{array}{rrrr

Dual description

- α_j is the price of one unit of nutrient j when it serves to reach the requirement; β_j what is paid not to exceed the cap
- λ_i and μ_i are the prices of the two semicontinuity constraints of food i, τ the price of variety
- The **objective** collects the requirements priced at α, pays the caps priced at β and collects the threshold t priced at τ
- **Columns of the x_i**: the nutritional content of one kilo of food i, priced at $\alpha_j - \beta_j$ and corrected by semicontinuity, cannot exceed the price w_i
- **Columns of the y_i**: switching on food i grants c_i units of minimum lot at price λ_i, forces at most d_i at price μ_i, and must cover the price τ of variety

A dual solution by hand (1/2)

- Set $\beta = \mu = \tau = 0$: maxima, caps and variety are not priced
- The columns of the y_i become $-c_i \, \lambda_i \leq 0$, satisfied with $\lambda = 0$
- What is left is $\sum_{j=1}^{r} g_{ij} \, \alpha_j \leq w_i$ for every food

The simplest recipe prices *a single* nutrient, at the lowest cost per gram among the foods:

$$\bar{\alpha}_j = \min_{i \,:\, g_{ij} > 0} \frac{w_i}{g_{ij}}, \qquad \text{bound} = a_j \, \bar{\alpha}_j$$

and one keeps the nutrient giving the larger bound.

A dual solution by hand (2/2)

On the instance:

$$\text{iron: } \bar{\alpha}_1 = \min\left(\tfrac{2}{10}, \tfrac{3}{20}, \tfrac{1}{5}, \tfrac{4}{25}\right) = \tfrac{3}{20}, \qquad 60 \cdot \tfrac{3}{20} = 9$$

$$\text{calcium: } \bar{\alpha}_2 = \min\left(\tfrac{2}{5}, \tfrac{3}{10}, \tfrac{1}{15}, \tfrac{4}{5}\right) = \tfrac{1}{15}, \qquad 40 \cdot \tfrac{1}{15} = \tfrac{8}{3}$$

The better one is the iron:

$$z(MILP) \geq LB = 9$$

Optimal solution

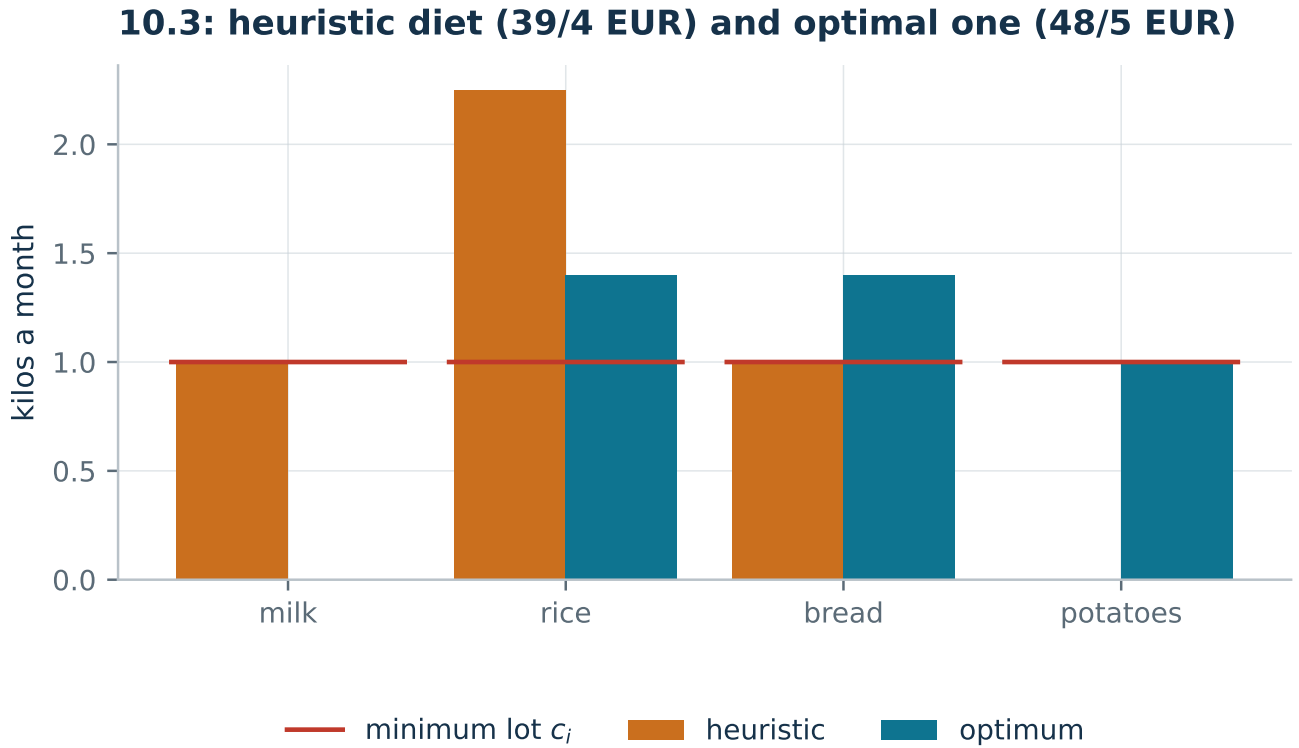
The optimum is obtained by solving the model: the two bounds just built are what helps the solver get there.

	milk	rice	bread	potatoes
x_i (kg)	0	1.4	1.4	1
y_i	0	1	1	1

Three different foods, 60 grams of iron and 40 of calcium: both exactly the minima.

$$z(MILP) = \frac{48}{5} = 9.6, \qquad 9 = LB \leq z(LP) = \frac{46}{5} \leq z(LP^+) = \frac{48}{5} = z(MILP) \leq UB = \frac{39}{4}$$

The optimal solution in a picture



Quantity per food in the heuristic diet and in the optimal one; the red bar marks the minimum lot c_i. The heuristic gap is a mere 1.6%: the tightest sandwich of the chapter.

Additional remarks

- The maximum constraints are not active at the optimum ($60 \leq 200$ and $40 \leq 150$): on this instance they could be removed, but on other instances they would bite
- The classic form of the diet problem (Stigler, 1945) has neither minimum lots nor counts: it is a pure LP. It is the two integer additions that turn it into a MILP
- The variety constraint can also be written as a *covering* constraint: $\sum_{i \in G_k} y_i \geq 1$ for every food group k. More informative from the nutritional point of view, and no harder
- Here $z(LP^+) = z(MILP)$ while $z(LP)$ is below: adding $y_i \leq 1$ does matter, because the caps $d_i = 8$ are much wider than the quantities in play

Modelling question: a higher minimum lot

The minimum lot goes up from 1 to 2 kilos for every chosen food.

How does the model change? What is the new optimum?

The final comparison

The three numbers that every problem of the course produces:

	value	what it is
UB	$\frac{39}{4}$	heuristic solution
LB	9	certified bound
$z(MILP)$	$\frac{48}{5}$	MILP optimum

- Gap of the heuristic: $\dfrac{\lvert UB - z(MILP)\rvert}{\lvert z(MILP)\rvert} = 1.6\%$
- Certified gap, known *before* solving the MILP (it cannot have $z(MILP)$ in the denominator): $UB - LB = \frac{3}{4}$, that is $\dfrac{UB - LB}{\lvert UB\rvert} = 7.7\%$

Worked variant: more variety (1/2)

At least four different foods are wanted instead of three: only the datum t changes, from 3 to 4.

- The optimum rises to $31/3 \approx 10.33$: one is forced to buy at least one kilo of the most expensive food (the potatoes), or to rearrange the quantities
- The cost of variety is therefore $31/3 - 48/5 = 11/15 \approx 0.73$ euros a month
- With $t = s$ the constraint becomes "all the foods" and the model stiffens until it coincides with the case in which all the indicators are one

Worked variant: more variety (2/2)

The count of the foods is not priced: with the recipe of the base problem ($\lambda = \mu = 0$ and a single positive α, on iron) it stays $\tau = 0$. The dual bound does not move, the optimum does.

	value	what it is
UB	$\frac{31}{3}$	heuristic solution
LB	9	dual certificate built by hand
$z(LP)$	$\frac{28}{3}$	relaxation without the bounds
$z(LP^+)$	$\frac{31}{3}$	relaxation with the bounds
$z(MILP)$	$\frac{31}{3}$	optimum of the MILP

The heuristic switches on the fourth food at its minimum lot, completes with bread and reaches 31/3, which is the optimum.

Summary: Diet with variety

- The semicontinuous variable comes from two constraints, minimum lot and activation: without the first one the count of the foods is empty
- Primal bound from the constructive heuristic: $UB = \frac{39}{4} = 9.75$
- Dual bound by hand pricing iron alone: $LB = 9$
- Optimum $z(MILP) = \frac{48}{5} = 9.6$, with $z(LP) = \frac{46}{5}$ and $z(LP^+) = \frac{48}{5}$: heuristic gap 1.6%, certified gap 7.7%