Relaxations, duality and bounds

Outline

What a relaxation is

The primal/dual conversion table

Weak duality, strong duality

Three recipes for building a dual solution by hand

A minimisation problem, in full

A maximisation problem: the roles swap

Valid inequalities and constraints that preserve optimality

Stronger formulations: what it means, and how it is proved

What the solver says

LP duals are not the marginal prices of the MILP

The bound protocol

Learning objectives (1/2)

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

- write the dual of the LP relaxation of a model with the primal/dual conversion table;
- build by hand a feasible dual solution and use its value as a bound on $z(MILP)$;
- tell a valid inequality from a constraint that only preserves optimality;

Learning objectives (2/2)

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

- compare two formulations by the strength of their LP relaxations;
- read `ObjVal`, `ObjBound` and `MIPGap` without mistaking LP duals for marginal prices of the MILP.

Prerequisites

- From this module, Chapter 1 «What is a MIP model»: the LP relaxation, the two kinds of bounds and the gap
- From the module *Linear Programming*, Chapter 9 «Duality theory»: the construction of the dual of a Linear Programming problem, weak and strong duality
- From the module *Integer Linear Programming*, Chapter 2 «Primal and dual bounds»: primal and dual bounds of an ILP problem

What a relaxation is

Introduction: What a relaxation is

- This chapter teaches how to produce, *by hand*, a number that certainly lies on one side of the integer optimum
- It serves three purposes:
  - understanding how good a model is
  - understanding how good a heuristic is
  - reading with judgement the numbers a solver reports when it has not finished
- We start from the definition of **relaxation** and from the three relaxations used in the course

Operational definition

- A **relaxation** of a minimisation problem with feasible set X is a problem with the same objective and a larger feasible set, $X \subseteq \hat{X}$
- The only thing relaxed are the constraints. Then

$$\min_{x \in \hat{X}} \sum_{j=1}^{n} c_j x_j \;\leq\; \min_{x \in X} \sum_{j=1}^{n} c_j x_j$$

  because the minimum over a larger set cannot be higher

- In a maximisation the inequality is reversed

The three relaxations of the course (1/2)

In increasing order of strength:

- $z(LP)$, **relaxation without the bounds**:
  - $x \in \{0, 1\}$ becomes $x \geq 0$
  - this is the one whose dual is written by hand: it has fewer constraints and so a dual with fewer variables
- $z(LP^+)$, **with the bounds**:
  - $x \in \{0, 1\}$ becomes $0 \leq x \leq 1$
  - this is Gurobi's `relax()`, the first relaxation the solver solves

The three relaxations of the course (2/2)

- $z(LP^{++})$, **with the bounds and the valid inequalities**:
  - as above, plus valid inequalities
  - the section on valid inequalities explains which ones

In a minimisation problem:

$$z(LP) \leq z(LP^{+}) \leq z(LP^{++}) \leq z(MILP)$$

The two relaxations coincide more often than one thinks

Observation 1: The two relaxations coincide more often than one thinks

If the model contains an assignment constraint $\sum_m x_{jm} = 1$ *with* $x \geq 0$, *then* $x_{jm} \leq 1$ *is already implied and the two relaxations are equal.*

- It happens whenever an equality constraint already forces the variables below one: this is the case of every assignment model
- When the two do differ, the difference must be reported: it is information about the model

Summary

Relaxation:

- Same objective, larger feasible set: in a minimisation it gives a value no higher than the optimum

Three relaxations:

- $z(LP)$ (without the bounds), $z(LP^+)$ (with the bounds), $z(LP^{++})$ (with the bounds and the valid inequalities)
- In a minimisation $z(LP) \leq z(LP^+) \leq z(LP^{++}) \leq z(MILP)$
- $z(LP)$ and $z(LP^+)$ coincide with assignment constraints

The primal/dual conversion table

Introduction: The primal/dual conversion table

- To build a bound by hand we need the **dual** of the relaxation
- The conversion table says how to write it: to every primal constraint corresponds a dual variable, to every primal variable a dual constraint
- We also see how the dual is **read**, with an economic meaning

From a minimisation primal to its dual

From a minimisation primal to its dual

Let the primal be a minimisation, with m constraints indexed by i and n variables indexed by j.

In the primal (min)	In the dual (max)
constraint i of type $\geq$	variable $\pi_i \geq 0$
constraint i of type $\leq$	variable $\pi_i \leq 0$
equality constraint i	free variable π_i
variable $x_j \geq 0$	constraint j of type $\leq c_j$
free variable x_j	equality constraint j, $= c_j$

The dual objective is $\max \sum_i b_i \pi_i$ and dual constraint j is $\sum_i a_{ij} \pi_i \lessgtr c_j$ with the direction given by the table. If the primal is a **maximisation**, every direction in the table is reversed and the dual is a minimisation.

From a minimisation primal to its dual (2/2)

- The dual objective is

$$\max \sum_i b_i \pi_i$$

- Dual constraint j is

$$\sum_i a_{ij} \pi_i \leqslant c_j$$

  with the direction given by the table

- If the primal is a **maximisation**, every direction in the table is reversed and the dual is a minimisation

How to read the dual

Dual constraint j says:

"the value I attach to the resources consumed by activity j cannot exceed its cost"

- With this reading, every recipe for building a feasible dual solution has an economic meaning
- That is what makes it memorable: we shall see three of them

Summary

Conversion table (minimisation primal):

- Constraint $\geq \rightarrow \pi_i \geq 0$; constraint $\leq \rightarrow \pi_i \leq 0$; equality $\rightarrow$ π_i free
- $x_j \geq 0 \rightarrow$ constraint $\leq c_j$; x_j free $\rightarrow$ constraint $= c_j$
- Maximisation primal: every direction is reversed

Economic reading:

- The resources consumed by an activity are worth no more than its cost

Weak duality, strong duality

Introduction: Weak duality, strong duality

- Two theorems connect the value of the primal and that of the dual
- **Weak duality** gives us a bound from *any* feasible dual solution
- **Strong duality** serves us as a check of the dual written by hand
- Finally we see why a bound of the LP is a bound for the MILP too

The two theorems, and what they are for here (1/2)

Let $(\bar{x}_1, \ldots, \bar{x}_n)$ be feasible for a minimisation LP and $(\bar{\pi}_1, \ldots, \bar{\pi}_m)$ feasible for its dual.

- **Weak duality**: *always*, with no assumptions,

$$\sum_{i=1}^{m} b_i \, \bar{\pi}_i \leq \sum_{j=1}^{n} c_j \, \bar{x}_j$$

  This is the one we need: it gives a lower bound from *any* feasible dual solution, even one built by hand and far from optimal

The two theorems, and what they are for here (2/2)

- **Strong duality**: if the LP has a finite optimum, then $z(D(LP)) = z(LP)$

  It serves as a *check*: the optimum of the dual written by hand must coincide with $z(LP)$, otherwise the dual is written wrong (the script verifies it with an `assert`)

- **The step to the integer problem**: every feasible solution of the MILP is feasible for the relaxation too, hence

$$\sum_{i=1}^{m} b_i \bar{\pi}_i \;\leq\; z(LP) \;\leq\; z(MILP)$$

- A dual bound of the LP is a bound for the MILP too, on the same side

There is no such thing as "the dual of the MILP"

Observation 2: There is no such thing as "the dual of the MILP"

The dual one writes is that of the relaxation, not of the MILP.

- A MILP has no linear dual
- Strong duality between a MILP and any linear program does not hold in general: the jump $z(MILP) - z(LP)$ is precisely what is missing
- When this course says "the dual", it always means the dual of the stated relaxation

Summary

Weak duality:

- Every feasible dual solution gives a bound, with no assumptions

Strong duality:

- $z(D(LP)) = z(LP)$: a check of the dual written by hand

For the MILP:

- $\sum_i b_i \bar{\pi}_i \leq z(LP) \leq z(MILP)$ (in a minimisation)
- The dual is always that of the relaxation

Three recipes for building a dual solution by hand

Introduction: Three recipes for building a dual solution by hand

- For weak duality *one* feasible dual solution is enough, not the optimal one
- We see three simple recipes to build it without a solver
- We shall use them in the examples of the next sections

The recipes (1/2)

1. **Zero out and saturate**
   - Set all dual variables to zero except one family
   - Push those to the largest value (in a maximisation dual) compatible with the dual constraints
   - Example, on an assignment: one sets $\bar{\pi} = 0$ and obtains $\bar{\mu}_j = \min_m c_{jm}$, that is "every job costs at least its cheapest option"

2. **Constructive heuristic on the constraints**
   - Scan the primal constraints one at a time
   - Raise the corresponding dual variable until the first dual constraint that opposes it becomes tight, and update the residuals

The recipes (2/2)

3. **The best ratio**
   - With a single capacity constraint in a maximisation, the dual variable
   $$\bar{v} = \max_j \frac{p_j}{w_j}$$
   is feasible, and gives the bound $b\bar{v}$
   - "All the capacity used at the best efficiency"

Whichever recipe is used, the constructed solution must be *checked* feasible for the dual (that is the only thing that makes the bound valid) and its value compared with $z(LP)$ to see how much was left on the table.

Summary

Three recipes:

- Zero out and saturate
- Constructive heuristic on the constraints
- The best ratio

Always:

- Check the dual feasibility of the constructed solution
- Compare its value with $z(LP)$

A minimisation problem, in full

Introduction: A minimisation problem, in full

- We apply everything to a minimum-cost **covering** problem
- We write the model, the dual of the relaxation and their versions on the instance
- We build by hand a lower bound (from the dual) and an upper bound (from a heuristic)
- We compare the sandwich obtained with the values of the solver

Minimum-cost zone covering

- A town has $n \in \mathbb{Z}_{\geq 1}$ districts, each with an emergency team
- Activating the team of district $j \in \{1, 2, \ldots, n\}$ costs $c_j \in \mathbb{Q}_{>0}$ thousand euros
- There are $m \in \mathbb{Z}_{\geq 1}$ sensitive zones: zone $i \in \{1, 2, \ldots, m\}$ is covered if at least one of the teams in the set S_i of the districts bordering it is active
- All zones must be covered at minimum cost

The data of the instance

$n = 4$ districts, of costs

$$c_1 = 4, \qquad c_2 = 3, \qquad c_3 = 5, \qquad c_4 = 3$$

and $m = 6$ zones, the six pairs of bordering districts:

zone i	1	2	3	4	5	6
S_i	$\{1,2\}$	$\{2,3\}$	$\{1,3\}$	$\{1,4\}$	$\{2,4\}$	$\{3,4\}$

S_i is the set of the teams covering zone i.

The variables

The following n binary variables are introduced:

$$x_j = \begin{cases} 1 & \text{if the team of district } j \text{ is activated} \\ 0 & \text{otherwise} \end{cases}$$

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

The model

Minimum-cost covering

$$\min \quad \sum_{j=1}^{n} c_j x_j \tag{2.1a}$$

$$\text{subject to} \quad \sum_{j \in S_i} x_j \geq 1, \qquad \forall i \in \{1, 2, \ldots, m\}, \tag{2.1b}$$

$$x_j \in \{0, 1\}, \quad \forall j \in \{1, 2, \ldots, n\}. \tag{2.1c}$$

Description of the model

- The **objective** sums the cost of the activated teams
- The m **covering constraints**, one per zone, say that at least one of the two teams bordering that zone is active: it is the structure of a *set covering*
- The **domains** are the n binarity conditions, one per team

The dual of the relaxation without the bounds

- The domain is relaxed to $x_j \geq 0$
- To each covering constraint we attach a dual variable $\pi_i \geq 0$ ($\geq$ direction in a minimisation, by the conversion table)

Dual of the LP relaxation

$$\max \quad \sum_{i=1}^{m} \pi_i \tag{2.2a}$$

$$\text{subject to} \quad \sum_{i\,:\,j \in S_i} \pi_i \leq c_j, \quad \forall j \in \{1, 2, \ldots, n\}, \tag{2.2b}$$

$$\pi_i \geq 0, \quad \forall i \in \{1, 2, \ldots, m\}. \tag{2.2c}$$

Description of the dual

- The variable π_i is the **price** attached to zone i
- The **objective** adds up the prices of the m zones
- The **constraints** are the columns of the x_j, one per team: the zones team j covers, priced at those values, cannot be worth more than what it costs to activate it, namely c_j
- This is the economic reading of the dual: no team may turn out to be a bargain

Covering on the instance: primal

Covering on the instance — primal

$$
\begin{aligned}
\min\ & 4x_1 + 3x_2 + 5x_3 + 3x_4 \\
& x_1 + x_2 \geq 1 \\
& x_2 + x_3 \geq 1 \\
& x_1 + x_3 \geq 1 \\
& x_1 + x_4 \geq 1 \\
& x_2 + x_4 \geq 1 \\
& x_3 + x_4 \geq 1 \\
& x_1, x_2, x_3, x_4 \in \{0, 1\}
\end{aligned}
$$

Covering on the instance: dual of the LP relaxation

Covering on the instance — dual of the LP relaxation

$$
\begin{aligned}
\max\ & \pi_1 + \pi_2 + \pi_3 + \pi_4 + \pi_5 + \pi_6 \\
& \pi_1 \qquad + \pi_3 + \pi_4 \qquad\qquad \le 4 \\
& \pi_1 + \pi_2 \qquad\qquad + \pi_5 \qquad \le 3 \\
& \qquad \pi_2 + \pi_3 \qquad\qquad + \pi_6 \le 5 \\
& \qquad\qquad\qquad \pi_4 + \pi_5 + \pi_6 \le 3 \\
& \pi_1, \quad \pi_2, \quad \pi_3, \quad \pi_4, \quad \pi_5, \quad \pi_6 \ge 0
\end{aligned}
$$

A dual solution by hand, recipe 2 (1/2)

Scan the zones in order; for zone i raise π_i by the most the residuals of the dual constraints of the teams in S_i allow, then update the residuals.

- **Zone 1** ($S_1 = \{1, 2\}$): residuals $(4, 3, 5, 3)$, the smallest among teams 1 and 2 is 3. Set $\bar{\pi}_1 = 3$; residuals $(1, 0, 5, 3)$
- **Zone 2** ($\{2, 3\}$): the residual of team 2 is 0, so $\bar{\pi}_2 = 0$
- **Zone 3** ($\{1, 3\}$): the smallest of 1 and 5 is 1. Set $\bar{\pi}_3 = 1$; residuals $(0, 0, 4, 3)$

A dual solution by hand, recipe 2 (2/2)

- **Zones 4 and 5** ($\{1,4\}$ and $\{2,4\}$): teams 1 and 2 have zero residual, so $\bar{\pi}_4 = \bar{\pi}_5 = 0$
- **Zone 6** ($\{3,4\}$): the smallest of 4 and 3 is 3. Set $\bar{\pi}_6 = 3$; residuals $(0,0,1,0)$

All dual constraints hold by construction, and the value is

$$LB \;=\; \bar{\pi}_1 + \cdots + \bar{\pi}_6 \;=\; 3+0+1+0+0+3 \;=\; 7$$

A primal upper bound

- The covering constructive heuristic picks, at each step, the team with the best ratio between cost and zones still uncovered
- It takes teams 1, 2 and 4, of cost

$$4 + 3 + 3 = 10$$

- The solution is feasible and *integer*, so

$$UB = 10$$

The sandwich, and what the solver says

- $z(LP) = z(LP^+) = 15/2$: the two relaxations coincide, and the optimum of the dual confirms it (strong duality)
- $z(MILP) = 10$, attained with teams $1, 2, 4$: the same solution as the heuristic, which here is optimal

UB (constructive heuristic)	10
LB (dual by hand)	7
$z(LP)$	15/2
$z(MILP)$	10
heuristic gap	0.0%

How the table is read

- The hand-built bound is 7, the relaxation 7.5, the optimum 10
- The certified gap between our two hand-built bounds is

$$\frac{10 - 7}{10} = 30\%$$

- Without solving the MILP we would know only that the optimum lies between 7 and 10
- The heuristic was already optimal, but the bounds cannot tell us that

Summary

Minimum-cost covering:

- Minimisation primal with $\geq$ constraints: maximisation dual with $\pi_i \geq 0$
- Dual by hand (recipe 2): $LB = 7$
- Constructive heuristic: $UB = 10$

The sandwich:

- $7 \leq z(LP) = 15/2 \leq z(MILP) = 10 \leq 10$
- Certified gap 30%, even though the heuristic is optimal

A maximisation problem: the roles swap

Introduction: A maximisation problem

- We repeat the same route on a **maximisation** problem: a knapsack with several resources
- In a maximisation the roles swap: the heuristic gives a **lower** bound, the dual an **upper** bound
- Finally we write the sandwich once and for all, for the minimisation and for the maximisation

Knapsack with several resources

- There are $n \in \mathbb{Z}_{\geq 1}$ items and $m \in \mathbb{Z}_{\geq 1}$ resources
- Item $j \in \{1, 2, \ldots, n\}$ has value $p_j \in \mathbb{Q}_{>0}$ and uses $a_{ij} \in \mathbb{Q}_{\geq 0}$ units of resource $i \in \{1, 2, \ldots, m\}$
- Of resource i there are $b_i \in \mathbb{Q}_{>0}$ units available
- Choose the subset of maximum value that fits inside *every* availability

The data of the instance

- $n = 5$ jobs of value $p = (10, 9, 7, 6, 4)$
- $m = 3$ resources (steel, lathe hours, testing hours), of availability $b = (10, 9, 8)$
- The consumptions, one row per resource and one column per job:

$$a = \begin{pmatrix} 6 & 4 & 3 & 3 & 2 \\ 1 & 3 & 4 & 2 & 3 \\ 3 & 2 & 2 & 4 & 1 \end{pmatrix}$$

The following n binary variables are introduced

$$x_j = \begin{cases} 1 & \text{if job } j \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \qquad \forall j \in \{1, 2, \ldots, n\}$$

The model

Knapsack with several resources

$$\max \quad \sum_{j=1}^{n} p_j x_j \tag{2.3a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \qquad \forall i \in \{1, 2, \ldots, m\}, \tag{2.3b}$$

$$x_j \in \{0, 1\}, \quad \forall j \in \{1, 2, \ldots, n\}. \tag{2.3c}$$

Description of the model

- The **objective** sums the value of the chosen jobs
- The **capacity constraints**, one per resource: the total consumption of the chosen jobs does not exceed the availability
- The **domains** are the n binarity conditions, one per job: a job is either done in full or not at all

The dual of the relaxation without the bounds

- The domain is relaxed to $x_j \geq 0$
- Maximisation primal with m constraints of type $\leq$: the table gives m dual variables $v_i \geq 0$, one per resource, and one dual constraint for each primal variable

Dual of the LP relaxation of the knapsack model

$$\min \quad \sum_{i=1}^{m} b_i \, v_i \tag{2.4a}$$

$$\text{subject to} \quad \sum_{i=1}^{m} a_{ij} \, v_i \geq p_j, \quad \forall j \in \{1, 2, \ldots, n\}, \tag{2.4b}$$

$$v_i \geq 0, \quad \forall i \in \{1, 2, \ldots, m\}. \tag{2.4c}$$

Description of the dual

- The variable v_i is the **price** of one unit of resource i
- The **objective** is the value attached to all the availabilities
- The n **constraints**, one per job: the resources used by job j, valued at those prices, are worth no less than the profit that job brings
- Every dual constraint now has m terms, one per resource: the column of x_j in the primal has m rows

Knapsack on the instance: primal

Knapsack on the instance — primal

$$
\begin{aligned}
\max\ & 10x_1 + 9x_2 + 7x_3 + 6x_4 + 4x_5 & \\
\text{subject to}\ & 6x_1 + 4x_2 + 3x_3 + 3x_4 + 2x_5 \leq & 10 \\
& x_1 + 3x_2 + 4x_3 + 2x_4 + 3x_5 \leq & 9 \\
& 3x_1 + 2x_2 + 2x_3 + 4x_4 + x_5 \leq & 8 \\
& x_1, \quad x_2, \quad x_3, \quad x_4, \quad x_5 \in \{0, 1\} &
\end{aligned}
$$

Knapsack on the instance: dual of the LP relaxation

Knapsack on the instance — dual of the LP relaxation

$$
\begin{aligned}
\min\ & 10v_1 + 9v_2 + 8v_3 \\
\text{subject to}\ & 6v_1 + v_2 + 3v_3 \geq 10 \\
& 4v_1 + 3v_2 + 2v_3 \geq 9 \\
& 3v_1 + 4v_2 + 2v_3 \geq 7 \\
& 3v_1 + 2v_2 + 4v_3 \geq 6 \\
& 2v_1 + 3v_2 + v_3 \geq 4 \\
& v_1, \quad v_2, \quad v_3 \geq 0
\end{aligned}
$$

The two bounds: the heuristic

- Constructive heuristic by ratio of value to total consumption
- The ratios are

$$\frac{10}{10} = 1, \quad \frac{9}{9} = 1, \quad \frac{7}{9}, \quad \frac{6}{9}, \quad \frac{4}{6}$$

- Job 1 is taken and then job 2; nothing else fits because the steel has run out
- Value $10 + 9 = 19$. In a **maximisation** the heuristic gives a **lower** bound:

$$LB = 19 \leq z(MILP)$$

The two bounds: the dual by hand

- Recipe 3, adapted: one pays *a single* resource
- Choosing resource i, set $\bar{v}_i = \max_j p_j / a_{ij}$ with all the others at zero: every dual constraint is satisfied by the single term $a_{ij}\bar{v}_i \geq p_j$
- The three choices give
  - steel: $10 \cdot \frac{7}{3} = \frac{70}{3}$
  - lathe: $9 \cdot 10 = 90$
  - testing: $8 \cdot \frac{9}{2} = 36$
- The cheapest is the steel. In a **maximisation** the dual gives an **upper** bound:

$$z(MILP) \leq UB = \frac{70}{3} \approx 23.33$$

The sandwich

$z(MILP) = 22$ (jobs 2, 3 and 4, which saturate all three resources), $z(LP) = 159/7 \approx 22.71$, $z(LP^+) = 22$:

$$19 \leq z(MILP) = 22 = z(LP^+) \leq z(LP) = \frac{159}{7} \leq UB = \frac{70}{3}$$

LB (constructive heuristic)	19
$z(MILP)$	22
$z(LP^+)$	22
$z(LP)$	159/7
UB (dual by hand)	70/3
heuristic gap	13.6%

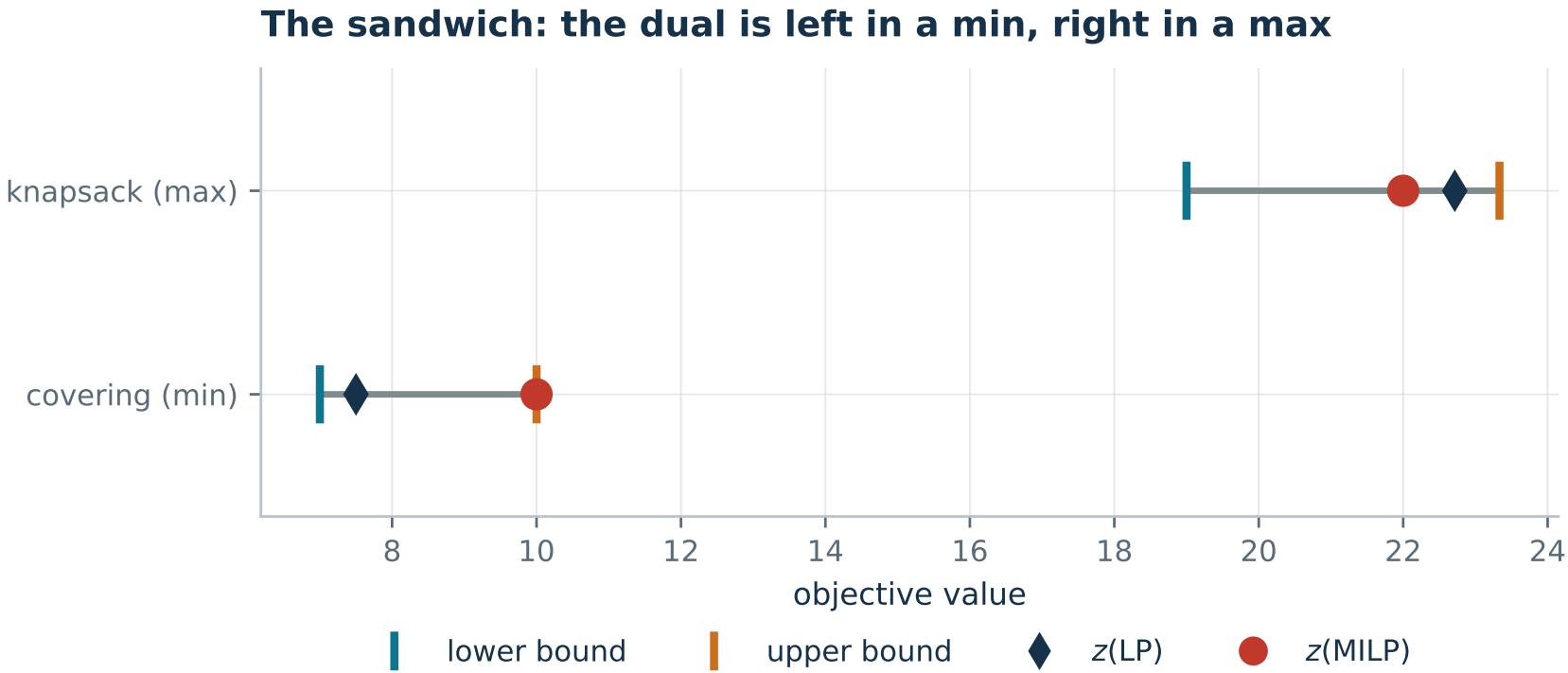
How the sandwich is read

- Here the dual by hand is *not* optimal for the relaxation:

$$\frac{70}{3} > \frac{159}{7}$$

- Paying a single resource is a valid recipe, but a coarse one: it ignores that the other two are saturated as well
- The relaxation with the bounds is worth 22 and coincides with the integer optimum
- The constraint $x_j \leq 1$ bites: without it the LP takes more than one unit of the best job

The two examples compared



In a minimisation the dual bound lies to the left of the optimum and the heuristic to the right; in a maximisation the roles swap.

The sandwich written once and for all (1/2)

Minimisation:

$$LB(\bar{\pi}) \leq z(D(LP)) = z(LP) \leq z(LP^+) \leq z(MILP) \leq UB(\bar{x})$$

Maximisation:

$$LB(\bar{x}) \leq z(MILP) \leq z(LP^+) \leq z(LP) = z(D(LP)) \leq UB(\bar{\pi})$$

where

- $(\bar{\pi}_1, \ldots, \bar{\pi}_m)$ is a feasible dual solution of the relaxation, of value $\sum_{i=1}^{m} b_i \bar{\pi}_i$
- $(\bar{x}_1, \ldots, \bar{x}_n)$ is a feasible solution of the MILP, of value $\sum_{j=1}^{n} c_j \bar{x}_j$

The sandwich written once and for all (2/2)

- The *relaxation side* is **optimistic** and holds all the dual bounds
- The *heuristic side* is **pessimistic** and holds all the feasible solutions
- The name (LB or UB) depends on the direction of the objective, the role does not

Summary

Knapsack with several resources:

- Maximisation primal with $\leq$ constraints: minimisation dual with $v_i \geq 0$
- Heuristic: $LB = 19$; dual by hand (a single resource): $UB = 70/3$
- $z(MILP) = z(LP^+) = 22$, $z(LP) = 159/7$

In general:

- Relaxation side optimistic, heuristic side pessimistic
- The name of the bound depends on the direction, the role does not

Valid inequalities and constraints that preserve optimality

Introduction: Valid inequalities

- Adding constraints can **tighten** the relaxation without changing the integer optimum
- We tell apart two different notions: **valid inequalities** and **constraints that preserve optimality**
- We see the **cover cut** for the knapsack, on an instance

Two different notions (1/2)

- A **valid inequality** for a MILP is satisfied by *all* feasible integer solutions
  - adding it does not change $z(MILP)$
  - it may reduce $z(LP^+)$, and then it is called a **cut**
- A **constraint that preserves optimality** cuts off some feasible solutions, but not all the optimal ones
  - after adding it an optimum still exists, and $z(MILP)$ does not change
  - it is not a valid inequality, and it must be declared as such

Two different notions (2/2)

Example:

- Let z be a variable the objective minimises, appearing only in constraints $z \geq t_j x_j$
- Adding $z \leq My$, with y an activation binary, cuts off feasible solutions (those with $y = 0$ and z positive)
- It does not cut off the optimal ones: at the optimum z is already pushed down by the objective

The difference matters:

- the first can always be added
- the second only after proving that at least one optimum survives, and never if one wants to enumerate *all* the optimal solutions

The cover cut for the knapsack

- A set S of items is a **cover** if

$$\sum_{j \in S} w_j > b$$

  that is, they do not all fit together
- Then

$$\sum_{j \in S} x_j \leq |S| - 1$$

  is a valid inequality
- The cover is **minimal** if no proper subset of it is already a cover: the cuts of minimal covers are the strongest

Example: minimal covers

Knapsack instance: $w = (5, 4, 3, 3)$, $b = 9$.

- The minimal covers are the four triples

$$\{1, 2, 3\}, \quad \{1, 2, 4\}, \quad \{1, 3, 4\}, \quad \{2, 3, 4\}$$

  of weight 12, 12, 11, 10: all > 9, and no pair is

- The four cuts are

$$\sum_{j \in S} x_j \leq 2$$

- The optimal solution of the relaxation, $z(LP^+) = 71/4$, is

$$\tilde{x} = (1,\ 1/4,\ 1,\ 0)$$

Example: the cuts on the solution of the relaxation

S	$\sum_{j \in S} \tilde{x}_j$	$\lvert S \rvert - 1$	
$\{1, 2, 3\}$	$9/4$	2	**violated**: the cut is needed
$\{1, 2, 4\}$	$5/4$	2	satisfied
$\{1, 3, 4\}$	2	2	satisfied (with equality)
$\{2, 3, 4\}$	$5/4$	2	satisfied

- With the four cuts $z(LP^+)$ drops from $71/4 = 17.75$ to $69/4 = 17.25$, and $z(MILP)$ stays 17
- The relaxation has moved closer to the integer optimum without changing it: this is the mechanism by which a modern solver closes gaps

Summary

Two notions:

- Valid inequality: it removes no integer solution; if it reduces $z(LP^+)$ it is a cut
- Constraint that preserves optimality: it removes feasible solutions but not all the optimal ones; it must be declared

Cover cut:

- $\sum_{j \in S} x_j \leq |S| - 1$ for every cover S
- On the instance: $z(LP^+)$ from 17.75 to 17.25, with $z(MILP) = 17$

Stronger formulations

Introduction: Stronger formulations

- The same problem can be written with different formulations
- We see the criterion for saying that one is **stronger** than another, and how it is proved
- And why stronger does not automatically mean faster

The criterion (1/2)

Two formulations A and B of the same problem are compared in two steps.

1. **Same integer set**:

$$X_A \cap \mathbb{Z}^n = X_B \cap \mathbb{Z}^n$$

   - without this step nothing is being compared
   - it is proved in both directions, using binarity or integrality where needed

2. **Relaxation**: B is *stronger* than A if $X_B \subseteq X_A$ as polyhedra, and then $z_B(LP^+)$ is at least as good as $z_A(LP^+)$
   - if the inclusion is strict *and* one exhibits a point of $X_A \setminus X_B$ optimal for the relaxation of A, the improvement is proved on that instance

The criterion (2/2): the activation

The reference case is the **activation**: the link between "customer i is served by site j" (x_{ij}) and "site j is open" (y_j).

- **Disaggregated**: $x_{ij} \leq y_j$ for every pair
- **Aggregated**: $\sum_i x_{ij} \leq n\, y_j$ for every site

- The first implies the second, so the disaggregated form is stronger
- On three customers and two sites the relaxation goes from 38/3 to 15, which is already the integer optimum

Stronger does not mean faster

Observation 3: Stronger does not mean faster

A stronger formulation has fewer nodes but more rows, and every node costs more. The balance depends on the instance.

What can be *proved* is the strength of the relaxation;

speed is *measured*.

Summary

Comparison between formulations:

- First: same integer set, proved in both directions
- Then: $X_B \subseteq X_A$ as polyhedra $\Rightarrow$ B stronger
- Activation: the disaggregated $x_{ij} \leq y_j$ is stronger than the aggregated one

Warning:

- Strength is proved, speed is measured

What the solver says

Introduction: What the solver says

- Gurobi reports several attributes at the end of a solve
- We see which ones to read, and in which **order**
- And why `ObjBound` must not be confused with the relaxation of the model as written

The four attributes to read, and in which order

1. `Status`: 2 is `OPTIMAL`, 3 `INFEASIBLE`, 9 `TIME_LIMIT`. To be read *before* anything else
2. `SolCount`: how many integer solutions were found. If it is 0, `ObjVal` does not exist and reading it is an error
3. `ObjVal` and `ObjBound`: the best solution found and the best bound proved. In a minimisation

$$\texttt{ObjBound} \leq z(MILP) \leq \texttt{ObjVal}$$

4. `MIPGap` and `NodeCount`: the certified gap between the two and how many nodes were explored

`ObjBound` is not the root relaxation (1/2)

Observation 4: `ObjBound` is not the root relaxation

On the covering instance the relaxation of the model as we wrote it is $15/2$ *and the integer optimum is* 10*. Yet Gurobi reports* `ObjBound` $= 10$ *and* `NodeCount` $= 0$*.*

- It closed the gap at the root, with presolve, its own cuts and heuristics, without ever branching
- Switching those off (`Presolve = Cuts = Heuristics = 0`) the same model gives the same optimum 10 but with 5 nodes

`ObjBound` is not the root relaxation (2/2)

Two consequences:

- "how hard a model is" is not a property of the model alone: it also depends on what the solver brings
- the relaxation we speak of in the hand-built bounds is that of the model as written, and it is obtained with `relax()`
- reading `ObjBound` at the end of the solve gives the optimum, not the relaxation

Summary

Order of reading:

- `Status`, then `SolCount`, then `ObjVal` and `ObjBound`, then `MIPGap` and `NodeCount`

`ObjBound`:

- It is the best bound proved by the solver, with its own tools
- It is not the relaxation of the model as written: for that one uses `relax()`

LP duals are not the marginal prices of the MILP

Introduction: LP duals and marginal prices

- In an LP the dual variables have an interpretation as **marginal prices** of the resources
- We show with a counterexample that for a MILP this interpretation is false
- And we see what, of the dual of the LP, remains true

The claim

- In an LP, the dual $\tilde{\pi}_i$ of a resource constraint is its **marginal price**
- Increasing b_i by a small amount changes the optimum by $\tilde{\pi}_i$ per unit (as long as the basis does not change)

For a MILP this is false,
and not by approximation.

The counterexample: the knapsack as the capacity varies

b	$z(MILP)$	$z(LP^+)$	LP dual	true change
8	16	16	2	—
9	17	71/4	7/4	+1
10	17	39/2	7/4	0
11	20	85/4	7/4	+3
12	23	23	7/4	+3

The last column is the true change of $z(MILP)$ with respect to the previous row.

How the counterexample is read

- The LP dual is the ratio p_j/w_j of the "critical" item:
  - it is 2 when the capacity runs out on item 1
  - it is $7/4$ when there is room left for item 2
- The true change of the integer optimum comes **in jumps**:
  - from $b = 9$ to $b = 10$ it does not change *at all* (no subset exploits the extra unit), while the dual promises $7/4$
  - from 10 to 11 it jumps by 3

What can be said, then

Observation 5: What can be said, then

Of the dual of the LP, the only use this course makes of it remains true: it is a ***bound****.*

- $\tilde{\pi}_i$ says how much the resource is worth *in the relaxation*, and $\sum_i b_i \tilde{\pi}_i$ is a valid bound for the MILP
- As managerial advice ("is it worth buying one more unit?") it must be checked by solving the MILP again
- The difference $z(MILP)(b_i + 1) - z(MILP)(b_i)$ is the only correct answer, and there is no closed formula for it

Summary

Marginal prices:

- In an LP the dual is the marginal price of the resource
- In a MILP the change of the optimum comes in jumps: the dual of the LP does not describe it

What remains:

- The dual of the LP gives a valid bound
- The value of one more unit is obtained only by solving the MILP again

The bound protocol

Introduction: The bound protocol

- Faced with any model, the route to tightening the sandwich is always the same
- It pays to follow it in order: six steps, from the two hand-built bounds to the additional remarks

The protocol (1/2)

1. a **feasible and integer solution**, built by hand, checked against every constraint *and* against integrality: it is the integrality check that makes that value a bound of the MILP and not of the relaxation alone
2. the **dual of the relaxation without the bounds**, written in general form and then on the instance, with the directions taken from the conversion table
3. a **feasible dual solution built by hand**, with the recipe stated, checked feasible and evaluated

The protocol (2/2)

4. the two **relaxations** $z(LP)$ and $z(LP^+)$, noting whether they coincide and why
5. the **optimum** $z(MILP)$ and the table $UB \cdot LB \cdot z(LP) \cdot z(LP^+) \cdot z(MILP)$ with the gap
6. the **additional remarks**: valid inequalities with their count, constraints preserving optimality declared as such, comparison of the relaxations of alternative formulations

The two sides of the sandwich

Steps 1 and 3 are the two sides of the sandwich,
and the only ones that can be built without a solver.

Steps 4 and 5 measure
how good the first two were.

Final summary (1/2)

- A **relaxation** enlarges the feasible set:
  $z(LP) \leq z(LP^{+}) \leq z(LP^{++}) \leq z(MILP)$ in a minimisation
- The **conversion table** gives the dual of the relaxation; **weak duality** makes every feasible dual solution a bound
- Three **recipes** to build a dual solution by hand: zero out and saturate, heuristic on the constraints, best ratio
- In a minimisation the dual gives LB and the heuristic UB; in a maximisation the roles swap

Final summary (2/2)

- **Valid inequalities** and cuts tighten the relaxation without changing the optimum
- A **stronger** formulation has a better relaxation, but it is not necessarily faster
- Of the **solver** one reads `Status`, `SolCount`, `ObjVal`/`ObjBound`, `MIPGap`; `ObjBound` is not the relaxation
- LP duals are **not** the marginal prices of the MILP, but they remain a bound
- The **bound protocol** puts everything in order, in six steps