

Packing problems

Outline

Why packing problems?

Overview of the family

The bin packing problem

The cutting stock problem

Remarks

Typical mistakes

In summary

Learning objectives

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

- ▶ state the definition of the bin packing problem and of the cutting stock problem, with instances and feasible solutions;
- ▶ explain the assumptions on the data;
- ▶ count the candidate solutions of a small instance and recognize that they are partitions;
- ▶ find a feasible solution by hand and compute the weight bound;
- ▶ use the two bounds to certify that a solution is optimal.

Prerequisites

- ▶ From this module, the chapter «[Combinatorial optimization problems](#)»: instances, feasible solutions, primal and dual bounds, the gap, and the Bell numbers that count the partitions of a set.
- ▶ From this module, the chapter «[Knapsack problems](#)»: the form of the definitions and of the assumptions on the data.

Why packing problems?

Introduction: Why packing problems?

- ▶ Before the formal definitions, we see where packing problems come from
- ▶ A story from logistics and computing, and a short history of the first results
- ▶ The cutting stock problem: the same problem with items in several identical copies
- ▶ A challenge on a small instance, which we solve in this chapter

A story

- ▶ Parcels of different weights must be shipped with identical trucks of capacity c .
- ▶ How many trucks are needed, and which parcels travel together?
- ▶ The same problem arises when virtual machines are placed on identical servers.



History

- ▶ The worst-case behaviour of simple packing rules, such as **First Fit**, was analysed in the early 1970s:
 - ▶ by Garey, Graham and Ullman (1972);
 - ▶ by Johnson, Demers, Ullman, Garey and Graham (1974).
- ▶ The latter proved that First Fit uses at most $\frac{17}{10}$ of the optimal number of bins, plus 2.
- ▶ We study First Fit and its guarantee later in the course.

Items in several copies

- ▶ The **cutting stock problem** is the same problem with items that come in several identical copies.
- ▶ The bins are rolls or sheets of material.
- ▶ The items are the pieces to cut.

A challenge

- ▶ Before reading on, try to answer two questions on the reference instance of the chapter: bins of capacity $c = 10$ and five items with weights

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
Weights	w_j	6	5	4	7	3

- ▶ How many bins are needed?
- ▶ How can we prove that fewer bins are not enough, without listing all the partitions?
- ▶ Here the second question has a simple answer.

Summary

Where packing problems come from:

- ▶ parcels on identical trucks, virtual machines on identical servers
- ▶ pieces cut from identical rolls or sheets: the cutting stock problem

History:

- ▶ simple packing rules, such as First Fit, analysed since the early 1970s

The challenge:

- ▶ how many bins, and how to prove that fewer bins are not enough

Overview of the family



Introduction: Overview of the family

- ▶ In this section we present the family of **packing problems**
- ▶ All its members share the same data and the same objective
- ▶ They differ only in the number of copies of each item that must be packed
- ▶ This common frame makes the two problems of the chapter easy to compare

The packing problems

- ▶ The **packing problems** share the same data: an infinite number of identical bins of *capacity* $c \in \mathbb{Z}_{\geq 1}$, a set $\mathcal{I} = \{1, 2, \dots, n\}$ of n items and, for each item $j \in \mathcal{I}$, a *weight* $w_j \in \mathbb{Z}_{\geq 1}$.
- ▶ A solution assigns every item, or every required copy of an item, to a bin: the total weight of the items in a bin must not exceed the capacity c .
- ▶ The objective is to minimize the number of bins used.

The family in one sentence

A packing problem places all the items into the fewest bins of capacity c .

The members of the family differ in the number of copies of each item that must be packed.

The members of the family

- The table lists the members of the family, with the section that defines each of them.

Problem	Acronym	Items to pack	Section
bin packing problem	BPP	one copy of each item	The bin packing problem
cutting stock problem	CSP	d_j copies of item j	The cutting stock problem

Summary

Common data:

- ▶ identical bins of capacity c , items $\mathcal{I} = \{1, 2, \dots, n\}$ with weights w_j

Common objective:

- ▶ pack everything without exceeding c in any bin, using the minimum number of bins

Two members:

- ▶ BPP: one copy of each item
- ▶ CSP: d_j copies of item j

The bin packing problem

Introduction: The bin packing problem

- ▶ We define the **bin packing problem** (BPP), its instances and its feasible solutions
- ▶ A feasible solution is a partition of the items into packing patterns, one per bin
- ▶ On the reference instance of the chapter we count the solutions, find a first solution by hand and compute a simple dual bound
- ▶ The two bounds together prove that the solution found by hand is optimal

The bin packing problem (BPP)

Definition 1: Bin Packing Problem

Given:

1. *an infinite number of bins, each with capacity $c \in \mathbb{Z}_{\geq 1}$,*
2. *a set $\mathcal{I} = \{1, 2, \dots, n\}$ of n items,*
3. *for each item $j \in \mathcal{I}$, a weight $w_j \in \mathbb{Z}_{\geq 1}$,*

*the **Bin Packing Problem** (BPP) asks to decide in which bin to place each item, so that the total weight of the items in any bin does not exceed the bin capacity, and the total number of bins used is minimized.*

BPP instance

► A **BPP instance** is specified by:

1. two integer numbers:

the bin capacity $c \in \mathbb{Z}_{\geq 1}$,

the number of items $n \in \mathbb{Z}_{\geq 1}$,

2. one vector of integer numbers of length n :

$\mathbf{w} \in \mathbb{Z}_{\geq 1}^n$, containing the item weights.

From the instance to the solutions

A solution says which items travel together:
one subset of items for each bin used.

These subsets are the **packing patterns**.

BPP feasible solution (1/2)

- A **BPP feasible solution** is a family of **packing patterns**, i.e., subsets of items each assigned to one bin,

$$\mathcal{P} \subseteq 2^{\mathcal{I}}$$

such that:

- (i) $\mathcal{P}$ forms a partition of $\mathcal{I}$ into *nonempty* packing patterns, namely

$$\begin{aligned} P &\neq \emptyset, \quad \text{for all } P \in \mathcal{P}, \\ P \cap P' &= \emptyset, \quad \text{for all } P, P' \in \mathcal{P} \text{ with } P \neq P', \\ \bigcup_{P \in \mathcal{P}} P &= \mathcal{I}; \end{aligned}$$

BPP feasible solution (2/2)

- (ii) the total weight of every packing pattern in $\mathcal{P}$ respects the bin capacity, namely

$$\sum_{j \in P} w_j \leq c, \quad \text{for all } P \in \mathcal{P}.$$

- Accordingly, the BPP can be equivalently stated as the problem of determining a partition of the item set into packing patterns such that the total weight of the items in each packing pattern does not exceed the bin capacity, and the number of packing patterns (bins) is minimized.

Formal statement of the BPP

Given a BPP instance, the BPP can be formally stated as:

$$\tilde{\mathcal{P}} \in \arg \min_{\mathcal{P} \subseteq 2^{\mathcal{I}}} \left\{ |\mathcal{P}| : \begin{array}{l} P \neq \emptyset, \quad \forall P \in \mathcal{P}, \\ P \cap P' = \emptyset, \quad \forall P, P' \in \mathcal{P}, P \neq P', \\ \bigcup_{P \in \mathcal{P}} P = \mathcal{I}, \\ \sum_{j \in P} w_j \leq c, \quad \forall P \in \mathcal{P} \end{array} \right\}.$$

Example (1/2)

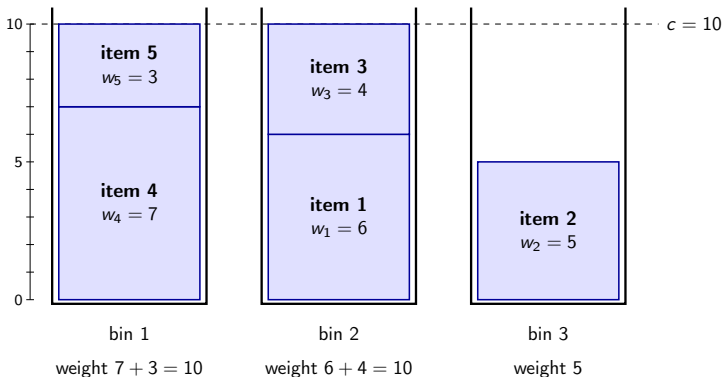
- ▶ We consider a bin of capacity $c = 10$ and a set $\mathcal{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following item weights:

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
Weights	w_j	6	5	4	7	3

- ▶ We call this BPP instance the **reference instance** of the chapter.

Example (2/2)

In the drawing every bin is a box of height $c = 10$ and every item a block of height w_j . The three bins hold $\{4, 5\}$, $\{1, 3\}$ and $\{2\}$; we show below that 3 bins are the minimum.



Assumptions

► We make the following assumptions:

1. The total weight of all items exceeds the bin capacity (otherwise, all items could fit in a single bin). Formally:

$$\sum_{j \in \mathcal{I}} w_j > c$$

2. Each item individually fits into a single bin (otherwise, it cannot be packed at all). Formally:

$$w_j \leq c, \quad \forall j \in \mathcal{I}$$

Counting the solutions

- ▶ **Candidate solutions.** The candidate solutions are the partitions of $\mathcal{I}$: they are counted by the Bell numbers (chapter «[Combinatorial optimization problems](#)»). For the reference instance we have $B(5) = 52$ partitions.
- ▶ **Feasible solutions.** A partition is feasible if every bin weighs at most 10. Enumeration gives 11 feasible partitions:
 - ▶ 1 with 5 bins (one item per bin),
 - ▶ 6 with 4 bins,
 - ▶ 4 with 3 bins.

Two bins are not enough, since the total weight 25 exceeds $2 \cdot 10$.

A first solution, by eye

- ▶ We pair the heavy items with the light ones:
 - ▶ item 4 (weight 7) with item 5 (weight 3),
 - ▶ item 1 (weight 6) with item 3 (weight 4);
 - ▶ item 2 (weight 5) stays alone.

- ▶ The solution

$$\mathcal{P} = \{ \{4, 5\}, \{1, 3\}, \{2\} \}$$

uses 3 bins, of weights 10, 10 and 5: it is a **primal bound**,

$$\text{UB} = 3 \geq z(I).$$

From a primal bound to a dual bound

The solution found by eye uses 3 bins.

Is 3 the minimum? We need a bound that holds for **every** feasible solution: a dual bound.

The weight bound

Observation 1: weight bound

Every feasible solution uses at least

$$\left\lceil \frac{\sum_{j \in \mathcal{J}} w_j}{c} \right\rceil$$

bins, so this value is a dual bound.

Proof

- ▶ If a solution uses k bins, the total weight it carries is at most $k c$.
- ▶ The total weight of the items is $\sum_j w_j$, hence

$$k c \geq \sum_j w_j.$$

- ▶ The number k is an integer, so

$$k \geq \left\lceil \frac{\sum_j w_j}{c} \right\rceil.$$



The dual bound and the gap

- ▶ In the reference instance we have $\sum_j w_j = 25$ and $c = 10$, so

$$\text{LB} = \lceil 25/10 \rceil = 3.$$

- ▶ The two bounds coincide: the gap is zero, the solution found by eye is optimal and $z(I) = 3$, without enumerating the partitions.
- ▶ The optimal solution is not unique: there are 4 optimal partitions,

$$\begin{aligned} & \{\{1\}, \{2, 3\}, \{4, 5\}\}, \quad \{\{2\}, \{1, 3\}, \{4, 5\}\}, \\ & \{\{4\}, \{1, 5\}, \{2, 3\}\}, \quad \{\{4\}, \{1, 3\}, \{2, 5\}\}. \end{aligned}$$

Summary (1/2)

The bin packing problem:

- ▶ an instance is c , n and the weights $\mathbf{w}$
- ▶ a feasible solution is a partition of $\mathcal{I}$ into nonempty packing patterns
- ▶ every packing pattern weighs at most c ; the number of patterns is minimized

Assumptions:

- ▶ $\sum_{j \in \mathcal{I}} w_j > c$ and $w_j \leq c$ for every item j

Summary (2/2)

On the reference instance:

- ▶ $B(5) = 52$ candidate solutions, 11 of them feasible
- ▶ a solution found by eye gives $UB = 3$

The weight bound:

- ▶ $\lceil \sum_{j \in \mathcal{J}} w_j / c \rceil$ is a dual bound, here $LB = 3$
- ▶ $LB = UB$: the gap is zero and $z(I) = 3$

The cutting stock problem

Introduction: The cutting stock problem

- ▶ We define the **cutting stock problem** (CSP): every item must be packed in a given number of identical copies
- ▶ The copies become the elements of an extended set of items, and a solution is again a partition
- ▶ On the reference CSP instance we repeat the steps of the BPP: we count the solutions, find one by eye and compute the bound
- ▶ The weight bound of the BPP still works, with the weights of all the copies

The cutting stock problem (CSP)

Definition 2: Cutting Stock Problem

Given:

1. *an infinite number of bins, each with capacity $c \in \mathbb{Z}_{\geq 1}$,*
2. *a set $\mathcal{J} = \{1, 2, \dots, n\}$ of n items,*
3. *for each item $j \in \mathcal{J}$, a weight $w_j \in \mathbb{Z}_{\geq 1}$ and a number of required copies $d_j \in \mathbb{Z}_{\geq 1}$,*

*the **Cutting Stock Problem** (CSP) asks to decide in which bin to place each copy of every item, so that the total weight of all item copies in any bin does not exceed the bin capacity, the required number of copies of each item is placed into the bins, and the total number of bins used is minimized.*

CSP instance

► A **CSP instance** is specified by:

1. two integer numbers:

the bin capacity $c \in \mathbb{Z}_{\geq 1}$,

the number of items $n \in \mathbb{Z}_{\geq 1}$,

2. two vectors of integer numbers of length n :

$\mathbf{w} \in \mathbb{Z}_{\geq 1}^n$, containing the item weights,

$\mathbf{d} \in \mathbb{Z}_{\geq 1}^n$, containing the required number of copies for each item.

Why «cutting stock»?

- ▶ The problem is called the *Cutting Stock Problem* because, in its classical application, the bins correspond to large stock rolls (or sheets of material).
- ▶ The rolls must be cut in order to produce the required copies of the items.
- ▶ The required copies are also known as **item demands**.

The extended set of items

- ▶ Given a CSP instance, we define the **extended set of items** as

$$\hat{\mathcal{J}} = \{ (j, k) : j \in \mathcal{J}, k \in \{1, 2, \dots, d_j\} \},$$

where each pair (j, k) represents the k -th required copy of item j .

- ▶ The cardinality of the extended set of items is

$$|\hat{\mathcal{J}}| = \sum_{j \in \mathcal{J}} d_j.$$

CSP feasible solution (1/2)

- A **CSP feasible solution** is a family of **packing patterns**, i.e., subsets of item copies each assigned to one bin,

$$\mathcal{P} \subseteq 2^{\hat{\mathcal{I}}}$$

such that:

- (i) $\mathcal{P}$ forms a partition of $\hat{\mathcal{I}}$ into *nonempty* packing patterns, namely

$$\begin{aligned} P &\neq \emptyset, \quad \text{for all } P \in \mathcal{P}, \\ P \cap P' &= \emptyset, \quad \text{for all } P, P' \in \mathcal{P} \text{ with } P \neq P', \\ \bigcup_{P \in \mathcal{P}} P &= \hat{\mathcal{I}}; \end{aligned}$$

CSP feasible solution (2/2)

- (ii) the total weight of the item copies in every packing pattern does not exceed the bin capacity, namely

$$\sum_{(j,k) \in P} w_j \leq c, \quad \text{for all } P \in \mathcal{P}.$$

- Accordingly, the CSP can be equivalently stated as the problem of determining a partition of the extended item set into packing patterns such that the total weight in each packing pattern does not exceed the bin capacity, and the number of packing patterns (bins) is minimized.

Formal statement of the CSP

Given a CSP instance, the CSP can be formally stated as:

$$\tilde{\mathcal{P}} \in \arg \min_{\mathcal{P} \subseteq 2^{\hat{\mathcal{J}}}} \left\{ |\mathcal{P}| : \begin{array}{l} P \neq \emptyset, \quad \forall P \in \mathcal{P}, \\ P \cap P' = \emptyset, \quad \forall P, P' \in \mathcal{P}, P \neq P', \\ \bigcup_{P \in \mathcal{P}} P = \hat{\mathcal{J}}, \\ \sum_{(j,k) \in P} w_j \leq c, \quad \forall P \in \mathcal{P} \end{array} \right\}.$$

Example (1/2)

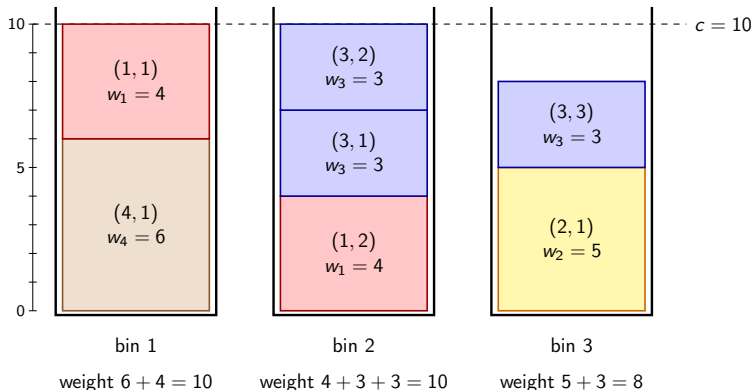
- ▶ We consider bins of capacity $c = 10$ and a set $\mathcal{J} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following weights and required copies:

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$
Weights	w_j	4	5	3	6
Copies	d_j	2	1	3	1

- ▶ We call this CSP instance the **reference CSP instance** of the chapter.

Example (2/2)

In the drawing the copies of the same item have the same colour, and the copy (j, k) is a block of height w_j . The 7 copies fit in 3 bins, of weights 10, 10 and 8.



Assumptions

► We make the following assumptions:

1. The total weight of all required item copies exceeds the bin capacity (otherwise, all copies could fit in a single bin).

Formally:

$$\sum_{j \in \mathcal{J}} d_j w_j > c.$$

2. Each item individually fits into a single bin (otherwise, no copy of that item can be packed at all). Formally:

$$w_j \leq c, \quad \forall j \in \mathcal{J}.$$

Counting the solutions

- ▶ **Candidate solutions.** The reference CSP instance has

$$2 + 1 + 3 + 1 = 7$$

copies.

- ▶ The partitions of the extended set of items are counted by

$$B(7) = 877,$$

where identical copies are counted as different items.

- ▶ 262 of them are feasible.

A first solution, by eye

- ▶ We fill three bins:
 - ▶ we pair item 4 (weight 6) with a copy of item 1 (weight 4),
 - ▶ we put the other copy of item 1 with two copies of item 3 ($4 + 3 + 3$),
 - ▶ and the last copy of item 3 with item 2 ($3 + 5$).
- ▶ The solution

$$\mathcal{P} = \{ \{(4, 1), (1, 1)\}, \{(1, 2), (3, 1), (3, 2)\}, \{(2, 1), (3, 3)\} \}$$

uses 3 bins, of weights 10, 10 and 8, hence

$$\text{UB} = 3.$$

The dual bound and the gap

- ▶ The weight bound of the bin packing problem holds with the weights of all the copies:

$$\text{LB} = \left\lceil \frac{\sum_j d_j w_j}{c} \right\rceil = \left\lceil \frac{28}{10} \right\rceil = 3.$$

- ▶ The gap is zero, so the solution is optimal and $z(I) = 3$.

The cutting stock problem:

- ▶ every item j is required in d_j identical copies
- ▶ a feasible solution is a partition of the extended set $\hat{\mathcal{I}}$ into packing patterns of weight at most c
- ▶ the BPP is the case $d_j = 1$ for every item

On the reference CSP instance:

- ▶ 7 copies, $B(7) = 877$ candidate solutions, 262 feasible
- ▶ by eye $UB = 3$, weight bound $LB = \lceil 28/10 \rceil = 3$: $z(I) = 3$

Remarks

Introduction: Remarks

- ▶ Some historical remarks on simple packing rules
- ▶ Simple rules, such as First Fit, give a solution quickly, but how far from the optimum?
- ▶ The first answers date back to the early 1970s

The worst case of First Fit

- ▶ The worst-case behaviour of simple packing rules, such as First Fit, was analysed by Garey, Graham and Ullman (1972) and by Johnson, Demers, Ullman, Garey and Graham (1974).
- ▶ Johnson et al. (1974) proved that First Fit uses at most $\frac{17}{10}$ of the optimal number of bins, plus 2.

Summary

Simple packing rules:

- ▶ their worst-case behaviour has been studied since the early 1970s

First Fit:

- ▶ at most $\frac{17}{10}$ of the optimal number of bins, plus 2 (Johnson et al., 1974)

Typical mistakes

Introduction: Typical mistakes

- ▶ We collect the most frequent mistakes on packing problems
- ▶ Two concern the bounds, two the way we count the solutions, one the assumptions on the data
- ▶ Recognizing them avoids wrong bounds and wrong counts

Typical mistakes (1/2)

Observation 2: the bounds

- ▶ ***Believing that the weight bound is always tight.***
With three items of weight 6 and $c = 10$ the bound gives $\lceil 18/10 \rceil = 2$, but no two items fit together, so 3 bins are needed.
- ▶ ***Counting the bins by the weights alone.*** *The weight bound ignores how the items fit together; it is a dual bound, and the primal bound still needs a solution.*

Typical mistakes (2/2)

Observation 3: the solutions and the data

- ▶ ***Listing the same packing twice.*** In the cutting stock problem the copies of an item are interchangeable: partitions that differ only by the exchange of identical copies are the same packing.
- ▶ ***Ignoring the order of the bins.*** A solution is a partition: $\{\{4, 5\}, \{1, 3\}, \{2\}\}$ and $\{\{2\}, \{4, 5\}, \{1, 3\}\}$ are the same solution.
- ▶ ***Forgetting the assumptions.*** An item with $w_j > c$ fits in no bin, and if the total weight is at most c one bin is enough.

Summary

Bounds:

- ▶ the weight bound is a dual bound, not always tight
- ▶ a primal bound always needs a feasible solution

Solutions:

- ▶ a solution is a partition: the order of the bins does not matter
- ▶ in the CSP, exchanging identical copies gives the same packing

Data:

- ▶ check $w_j \leq c$ and $\sum_j w_j > c$

In summary

Introduction: In summary

- ▶ We collect the key ideas of the chapter
- ▶ The two problems, the structure of their solutions, and the two bounds
- ▶ A dashboard with the bounds of the two reference instances

Summary (1/2)

- ▶ The BPP partitions the items into the minimum number of bins of capacity c ; the cutting stock problem is the BPP with several identical copies of every item, and the BPP is the case $d_j = 1$.
- ▶ A solution is a partition, so the candidates are counted by the Bell numbers and grow very fast.
- ▶ A solution found by eye is a primal bound; the weight bound $\lceil \sum_j w_j / c \rceil$ is a dual bound.
- ▶ When the two bounds coincide, the gap is zero and the solution is certified optimal.

Summary (2/2)

Dashboard of the reference instances.

instance (min)	lower bound	upper bound	from
BPP, reference instance	3 (weight bound)	3 (by eye)	this chapter
CSP, reference CSP instance	3 (weight bound)	3 (by eye)	this chapter