

Knapsack problems

Outline

Why knapsack problems?

Overview of the family

The 0/1 knapsack problem

The subset sum problem

The bounded knapsack problem

The unbounded knapsack problem

The multidimensional knapsack problem

The multiple knapsack problem

The multiple-choice knapsack problem

The knapsack problem with setups

Remarks

Typical mistakes

In summary

Learning objectives

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

- ▶ state the definition of the 0/1 knapsack problem and of its variants, with instances and feasible solutions;
- ▶ explain the assumptions on the data and why they can be made without loss of generality;
- ▶ count the candidate solutions and the feasible solutions of a small instance;
- ▶ find a feasible solution by hand and compute a simple combinatorial dual bound;
- ▶ tell the variants apart: copies of the items, several resources, several knapsacks, classes of items.

Prerequisites

- ▶ From this module, the chapter «[Combinatorial optimization problems](#)»: instances, feasible solutions, primal and dual bounds and the gap.
- ▶ From the *Preliminaries*: subsets of a finite set and the number 2^n of subsets.

Why knapsack problems?

Introduction: why knapsack problems?

- ▶ Before the definitions, we see where knapsack problems come from: a story and some applications
- ▶ Choosing the most valuable items under a limited capacity is one of the most common decisions in practice
- ▶ A bit of history: the problem was studied in the early days of integer programming
- ▶ We end with a challenge on the reference instance of the chapter

A story

- ▶ A fund has a budget of c euros and n candidate projects
- ▶ Project j costs w_j euros and brings a profit p_j
- ▶ **Which projects should be financed to maximize the total profit?**



The same structure, everywhere

The same structure appears when we:

- ▶ load the most valuable cargo into a truck;
- ▶ choose the advertisements for a limited space;
- ▶ select the investments of a limited budget.

In every case:

- ▶ a **capacity** c (budget, load, space);
- ▶ **items** j , each with a **weight** w_j (what it uses) and a **profit** p_j (what it brings).

History

- ▶ The problem was studied in the early days of integer programming.
- ▶ **Dantzig (1957)** sorted the items by non-increasing efficiency p_j/w_j and used this order to solve the continuous version, an idea that returns in the bounds of the course.
- ▶ **Karp (1972)** put the subset sum problem, called knapsack in his paper, among the first 21 NP-complete problems. The KP contains the SSP as the special case $p_j = w_j$, so it is NP-hard.
- ▶ **Ibarra and Kim (1975)** gave the first fully polynomial-time approximation scheme for the knapsack problem (later in the course).

A challenge

Before reading on, try to answer two questions on the reference instance of the chapter: capacity $c = 9$ and 6 items.

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
Profits	p_j	6	5	8	9	6	7
Weights	w_j	2	3	6	7	5	9

1. How many subsets of items are feasible, and which one is best?
2. How can we be sure that nothing better exists, without listing all the subsets?

The chapter answers the first question and gives a first bound for the second.

Where knapsack problems come from:

- ▶ Select items of maximum total profit within a capacity:
projects within a budget, cargo in a truck, advertisements in a space, investments

History:

- ▶ Dantzig (1957): the order of efficiency p_j/w_j
- ▶ Karp (1972): the subset sum problem is NP-complete, so the KP is NP-hard
- ▶ Ibarra and Kim (1975): the first fully polynomial-time approximation scheme

Overview of the family

Introduction: overview of the family

- ▶ In this section we meet the whole family of knapsack problems
- ▶ All the members share the same data: items with weights and profits, knapsacks with capacities
- ▶ They differ in a few features: copies of the items, number of resources, number of knapsacks, classes of items
- ▶ This gives the map of the chapter: each member is then defined in its own section

The common data

- ▶ The **knapsack problems** share the same data:
 - ▶ a set $\mathcal{I} = \{1, 2, \dots, n\}$ of n items,
 - ▶ for each item $j \in \mathcal{I}$ a *weight* $w_j \in \mathbb{Z}_{\geq 1}$ and a *profit* $p_j \in \mathbb{Z}_{\geq 1}$,
 - ▶ one or more knapsacks with a *capacity* in $\mathbb{Z}_{\geq 1}$.
- ▶ A solution decides which items are selected, or how many copies of each item: the total weight of the selected items must not exceed the capacity of the knapsack that receives them.
- ▶ The objective is to maximize the total profit of the selected items.

The family in one sentence

A knapsack problem selects items of maximum total profit whose total weight respects the capacities.

The members of the family differ in the number of copies of the items, in the number of capacities and of knapsacks, and in the classes into which the items are grouped.

The members of the family (1/2)

The table lists the members of the family; each one is defined in its own section.

Problem	Acronym	Difference with respect to the KP
0/1 knapsack problem	KP	one copy of each item, one knapsack
subset sum problem	SSP	$p_j = w_j$, total weight exactly equal to c
bounded knapsack problem	BKP	at most b_j copies of item j
unbounded knapsack problem	UKP	an unlimited number of copies of each item

The members of the family (2/2)

Problem	Acronym	Difference with respect to the KP
multidimensional knapsack problem	MDKP	k capacities c_d and weights w_{dj}
multiple knapsack problem	MKP	m knapsacks with capacities c_k
multiple-choice knapsack problem	MCKP	classes $\mathcal{J}_h$, exactly one item per class
knapsack problem with setups	KPS	classes $\mathcal{J}_h$ with setup costs f_h and setup weights s_h

The variants at a glance

- ▶ Each problem of the family is illustrated on a small instance, in its section and with the notation of that section.
- ▶ The next tables list, for these instances, the number of candidate solutions, the number of feasible solutions and the optimal value, computed by enumeration.
- ▶ In the BKP and in the UKP a candidate solution is the vector of the numbers of copies of every item, so that identical copies are not counted twice.
- ▶ The optimal solution is not always unique: the MKP, the MCKP and the KPS have several optimal solutions with the same value, and in the SSP three subsets reach the target.

The variants at a glance: candidate and feasible solutions

Variant	Candidate solutions	Number	Feasible
KP	subsets of 6 items	64	13
SSP	subsets of 6 items	64	3
BKP	copies $x_j \leq b_j$	144	18
UKP	copies $x_j \leq \hat{b}_j$	320	22
MDKP	subsets of 4 items	16	9
MKP	each item in a knapsack or none	243	109
MCKP	one item per class	8	3
KPS	subsets of 6 items	64	40

The variants at a glance: optimal solutions

Variant	Optimum	Optimal solutions
KP	15	$\{1, 4\}$
SSP	yes	$\{6\}, \{1, 4\}, \{2, 3\}$
BKP	18	two copies of item 1 and item 5
UKP	24	four copies of item 1
MDKP	14	$\{3, 4\}$
MKP	30	$(S_1, S_2) = (\{1, 2\}, \{4, 5\}),$ $(\{2, 4\}, \{1, 5\}), (\{4, 5\}, \{1, 2\})$
MCKP	17	$\{1, 4, 5\}, \{2, 3, 5\}$
KPS	16	$\{1, 2, 3\}, \{4, 5, 6\}, \{1, 2, 3, 6\}$

In the MCKP only 3 of the 8 choices of one item per class fit in the knapsack: the feasibility condition among the assumptions of the MCKP is not a formality.

Summary

The family:

- ▶ Same data: items with weights w_j and profits p_j , one or more knapsacks with capacities
- ▶ Same goal: maximize the total profit of the selected items within the capacities

The members:

- ▶ KP, SSP, BKP, UKP, MDKP, MKP, MCKP, KPS
- ▶ They differ in copies of the items, resources, knapsacks and classes of items
- ▶ On the small instances the optimal solution is not always unique

The 0/1 knapsack problem

Introduction: the 0/1 knapsack problem

- ▶ We define the 0/1 knapsack problem (KP), the basic member of the family
- ▶ Each item is either selected (once) or not selected: a solution is a subset of items
- ▶ We introduce the reference instance of the chapter
- ▶ On it we count the solutions, find a first solution by eye and compute a simple dual bound

Definition 1: 0/1 Knapsack Problem

Given:

1. a knapsack of 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 profit $p_j \in \mathbb{Z}_{\geq 1}$ and a weight $w_j \in \mathbb{Z}_{\geq 1}$,

the 0/1 Knapsack Problem (KP) asks to decide which items to insert into the knapsack, so that the total weight of the selected items does not exceed the knapsack capacity, and the total profit of the selected items is maximized.

KP instance

A **KP instance** is specified by:

1. two integer numbers:
the number of items $n \in \mathbb{Z}_{\geq 1}$,
the knapsack capacity $c \in \mathbb{Z}_{\geq 1}$,
2. two vectors of integer numbers of length n :
 $\mathbf{p} \in \mathbb{Z}_{\geq 1}^n$, containing the item profits,
 $\mathbf{w} \in \mathbb{Z}_{\geq 1}^n$, containing the item weights.

KP feasible solution

A **KP feasible solution** is a subset of items

$$S \subseteq \mathcal{I}$$

such that the total weight of the selected items does not exceed the knapsack capacity, namely

$$\sum_{j \in S} w_j \leq c.$$

Accordingly, the KP can be equivalently stated as the problem of selecting a subset of items whose total weight respects the knapsack capacity, and whose total profit is maximized.

The KP in formal terms

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

$$\tilde{S} \in \arg \max_{S \subseteq \mathcal{I}} \left\{ \sum_{j \in S} p_j : \sum_{j \in S} w_j \leq c \right\}.$$

We now look at a KP instance with 6 items:
it is the **reference instance** of the chapter.

Example (1/2)

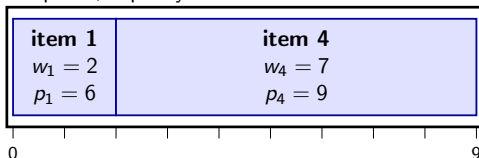
We consider a knapsack of capacity $c = 9$ and a set $\mathcal{I} = \{1, 2, 3, 4, 5, 6\}$ of 6 items ($n = 6$), with the following profits and weights:

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
Profits	p_j	6	5	8	9	6	7
Weights	w_j	2	3	6	7	5	9

- ▶ In the drawing, the knapsack is a box of length $c = 9$ and every item is a block of length w_j .
- ▶ The items 1 and 4 fill the knapsack, with weight 9 and profit 15; the other items stay out (dashed).
- ▶ This is the reference instance of the chapter.

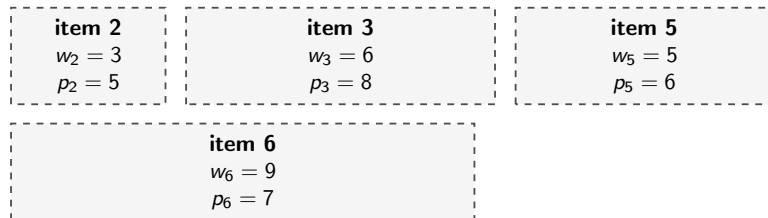
Example (2/2)

knapsack, capacity $c = 9$



weight $2 + 7 = 9$
profit $6 + 9 = 15$

not selected



Assumptions

We make the following assumptions:

1. The total weight of all items exceeds the knapsack capacity (otherwise, all items could be selected). Formally:

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

2. Each item individually fits into the knapsack (otherwise, it cannot be selected). Formally:

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

How many solutions does the reference instance have?
And how good is a solution found by hand?

Counting the solutions

- ▶ **Candidate solutions.** Every subset of $\mathcal{I}$ is a candidate solution, so there are 2^n of them (see «[Combinatorial optimization problems](#)»). For the reference instance we have $2^6 = 64$.
- ▶ **Feasible solutions.** The three lightest items weigh $2 + 3 + 5 = 10 > 9$, so no subset with three or more items is feasible. The feasible solutions are the empty set, the 6 singletons and the 6 pairs of the following table: 13 solutions out of 64.

pair S	$\{1, 2\}$	$\{1, 3\}$	$\{1, 4\}$	$\{1, 5\}$	$\{2, 3\}$	$\{2, 5\}$
weight	5	8	9	7	9	8
profit	11	14	15	12	13	11

A first solution, by eye

- ▶ Item 4 has the largest profit (9, weight 7).
- ▶ The residual capacity $9 - 7 = 2$ is enough for item 1 (weight 2, profit 6).
- ▶ The subset $\{1, 4\}$ has weight 9 and profit 15: it is a **primal bound**,

$$\text{LB} = 15 \leq z(I).$$

A feasible solution bounds $z(I)$ from below.
To bound $z(I)$ from above, we count how many items fit at most.

Observation 1: cardinality bound

Let $\bar{k}$ be the largest integer such that the $\bar{k}$ lightest items weigh at most c . Every feasible subset has at most $\bar{k}$ items, hence $z(I)$ is at most the sum of the $\bar{k}$ largest profits.

Proof

- ▶ A subset with k items weighs at least as much as the k lightest items.
- ▶ If $k > \bar{k}$, the k lightest items weigh more than c , so the subset is not feasible.
- ▶ The profit of a subset with at most $\bar{k}$ items is at most the sum of the $\bar{k}$ largest profits.



A first dual bound

- ▶ In the reference instance we have $\bar{k} = 2$, since

$$2 + 3 = 5 \leq 9 \quad \text{and} \quad 2 + 3 + 5 = 10 > 9.$$

- ▶ The two largest profits are 9 and 8, so

$$\text{UB} = 9 + 8 = 17.$$

- ▶ The sum of all the profits, 41, is a much weaker dual bound.

The gap

- ▶ We have

$$15 \leq z(I) \leq 17 :$$

the absolute gap is 2 and the relative gap is $\frac{2}{15}$.

- ▶ Listing the 13 feasible subsets shows that $\{1, 4\}$ is the unique optimal solution and $z(I) = 15$.
- ▶ For larger instances we need stronger bounds, which come from relaxations (module *Integer linear programming*, «[Primal and dual bounds](#)»).

Summary

The 0/1 knapsack problem:

- ▶ One copy of each item, one knapsack: a solution is a subset $S \subseteq \mathcal{I}$ with $\sum_{j \in S} w_j \leq c$
- ▶ Assumptions: not all the items fit, each item fits alone
- ▶ There are 2^n candidate solutions

On the reference instance:

- ▶ 13 feasible subsets out of 64
- ▶ $LB = 15$ by eye, $UB = 17$ from the cardinality bound
- ▶ $\{1, 4\}$ is the unique optimal solution, $z(I) = 15$

The subset sum problem

Introduction: the subset sum problem

- ▶ The subset sum problem (SSP) asks to reach a target value *exactly* with the weights of a subset of items
- ▶ Its answer is yes or no: it is not an optimization problem
- ▶ It can be solved as a KP, by setting the profits equal to the weights
- ▶ It is the special case that makes the KP NP-hard

Definition 2: Subset Sum Problem

Given:

1. a target value $c \in \mathbb{Z}_{\geq 1}$,
2. a set $\mathcal{J} = \{1, 2, \dots, n\}$ of n items,
3. for each item $j \in \mathcal{J}$, a weight $w_j \in \mathbb{Z}_{\geq 1}$,

the **Subset Sum Problem (SSP)** asks whether there exists a subset of items whose total weight is exactly equal to the target value c .

SSP instance

An **SSP instance** is specified by:

1. two integer numbers:
 - the number of items $n \in \mathbb{Z}_{\geq 1}$,
 - the target value $c \in \mathbb{Z}_{\geq 1}$,
2. one vector of integer numbers of length n :
 - $\mathbf{w} \in \mathbb{Z}_{\geq 1}^n$, containing the item weights.

SSP feasible solution

An **SSP feasible solution** is a subset of items

$$S \subseteq \mathcal{I}$$

such that the total weight of the selected items is exactly equal to the target value, namely

$$\sum_{j \in S} w_j = c.$$

Accordingly, the SSP can be equivalently stated as the problem of selecting a subset of items whose total weight matches the target value exactly.

The SSP in formal terms

Given an SSP instance, the SSP can be formally stated as:

$$\text{find } S \subseteq \mathcal{J} \quad \text{such that} \quad \sum_{j \in S} w_j = c.$$

Example (1/2)

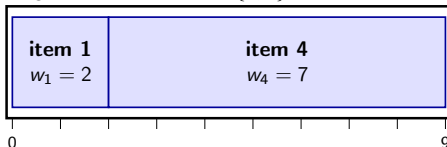
We consider a target value $c = 9$ and a set $\mathcal{J} = \{1, 2, 3, 4, 5, 6\}$ of 6 items ($n = 6$), with the following weights:

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
Weights	w_j	2	3	6	7	5	9

- ▶ The subset $S = \{1, 4\}$ is feasible since $w_1 + w_4 = 2 + 7 = 9$.
- ▶ The subset $S = \{6\}$ is feasible since $w_6 = 9$.
- ▶ The drawing shows the three subsets whose weights fill the box of length $c = 9$ exactly: $\{1, 4\}$, $\{6\}$ and $\{2, 3\}$.

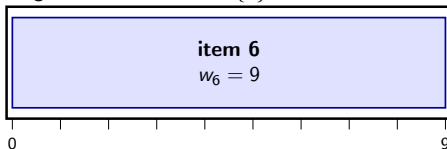
Example (2/2)

target $c = 9$, subset $S = \{1, 4\}$



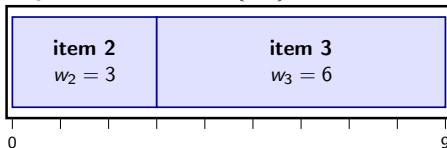
$$2 + 7 = 9$$

target $c = 9$, subset $S = \{6\}$



$$9 = 9$$

target $c = 9$, subset $S = \{2, 3\}$



$$3 + 6 = 9$$

Solving the SSP with the KP

- ▶ The SSP is not an **optimization problem**, since its answer is either **yes** or **no**. However, it can be solved by using an optimization problem.
- ▶ An SSP instance can be solved as a KP instance by setting the profit of each item equal to its weight, namely

$$p_j = w_j, \quad \forall j \in \mathcal{I}.$$

Solving the SSP with the KP: the answer

In this case, we solve the following KP:

$$\tilde{S} \in \arg \max_{S \subseteq \mathcal{I}} \left\{ \sum_{j \in S} w_j \quad : \quad \sum_{j \in S} w_j \leq c \right\}.$$

- ▶ If $\sum_{j \in \tilde{S}} w_j = c$, then the answer to the SSP is **yes**.
- ▶ Otherwise, if $\sum_{j \in \tilde{S}} w_j < c$, the answer is **no**.

Assumptions

We make the following assumptions:

1. The total weight of all items is at least the target value (otherwise, no feasible subset can exist). Formally:

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

2. Each item weight is not larger than the target value (otherwise, it cannot belong to any feasible subset). Formally:

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

Summary

The subset sum problem:

- ▶ Find a subset of items whose total weight is exactly equal to the target value c
- ▶ The answer is yes or no: it is not an optimization problem

Solving it with the KP:

- ▶ Set $p_j = w_j$ and solve the KP: the answer is yes if and only if the optimal subset weighs exactly c
- ▶ In the example, three subsets reach the target $c = 9$: $\{1, 4\}$, $\{6\}$, $\{2, 3\}$

The bounded knapsack problem

Introduction: the bounded knapsack problem

- ▶ In the bounded knapsack problem (BKP) each item is available in several copies, at most b_j
- ▶ The decision is how many copies of each item to insert
- ▶ We turn it into the selection of a subset of an **extended set of items**, one element per copy
- ▶ The same idea returns in the unbounded knapsack problem

Definition 3: Bounded Knapsack Problem

Given:

1. a knapsack of 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 profit $p_j \in \mathbb{Z}_{\geq 1}$, a weight $w_j \in \mathbb{Z}_{\geq 1}$, and a number of available copies $b_j \in \mathbb{Z}_{\geq 1}$,

the **Bounded Knapsack Problem (BKP)** asks to decide how many copies of each item to insert into the knapsack, so that the total weight of all selected item copies does not exceed the knapsack capacity, each item is taken no more than its available copies, and the total profit of all selected item copies is maximized.

BKP instance

A **BKP instance** is specified by:

1. two integer numbers:

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

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

2. three vectors of integer numbers of length n :

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

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

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

Extended set of items

Given a BKP instance, we define the **extended set of items** as

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

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

The cardinality of the extended set of items is

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

BKP feasible solution

A **BKP feasible solution** is a subset of items

$$S \subseteq \hat{\mathcal{I}}$$

such that the total weight of the selected items does not exceed the knapsack capacity, namely

$$\sum_{(j,k) \in S} w_j \leq c.$$

Accordingly, the BKP can be equivalently stated as the problem of selecting a subset of the extended set of items whose total weight respects the knapsack capacity, and whose total profit is maximized.

The BKP in formal terms

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

$$\tilde{S} \in \arg \max_{S \subseteq \hat{\mathcal{J}}} \left\{ \sum_{(j,k) \in S} p_j : \sum_{(j,k) \in S} w_j \leq c \right\}.$$

Example (1/3)

We consider a knapsack of capacity $c = 9$ and a set $\mathcal{I} = \{1, 2, 3, 4, 5, 6\}$ of 6 items ($n = 6$), with the following profits, weights, and available copies:

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
Profits	p_j	6	5	8	9	6	7
Weights	w_j	2	3	6	7	5	9
Available copies	b_j	2	2	1	1	1	1

The extended set of items is:

$$\hat{\mathcal{I}} = \{ (1, 1), (1, 2), (2, 1), (2, 2), \\ (3, 1), (4, 1), (5, 1), (6, 1) \}.$$

Example (2/3)

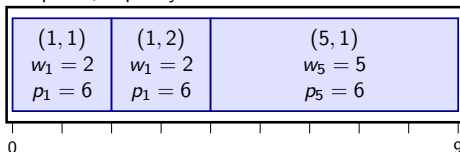
Each copy (j, k) has profit p_j and weight w_j .

Copy label	1	2	3	4	5	6	7	8
(j, k)	(1, 1)	(1, 2)	(2, 1)	(2, 2)	(3, 1)	(4, 1)	(5, 1)	(6, 1)
Profit p_j	6	6	5	5	8	9	6	7
Weight w_j	2	2	3	3	6	7	5	9

The drawing on the next slide shows an optimal solution: the two copies of item 1 and the copy of item 5, with weight $2 + 2 + 5 = 9$ and profit $6 + 6 + 6 = 18$.

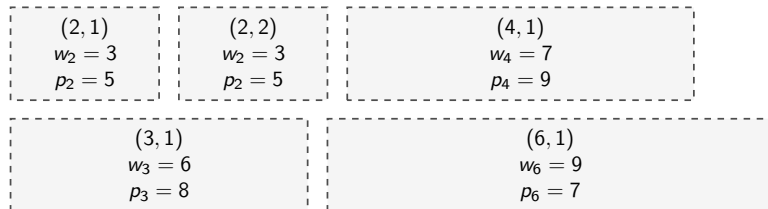
Example (3/3)

knapsack, capacity $c = 9$



weight $2 + 2 + 5 = 9$
profit $6 + 6 + 6 = 18$

copies not selected



Assumptions (1/2)

We make the following assumptions:

1. The total weight of all available copies of all items exceeds the knapsack capacity (otherwise, all copies could be selected).

Formally:

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

2. Each item individually fits into the knapsack (otherwise, it cannot be selected). Formally:

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

Assumptions (2/2)

3. The total weight of all available copies of any single item does not exceed the knapsack capacity (otherwise, that item could never be taken in the maximum allowed number of copies).

Formally:

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

Summary

The bounded knapsack problem:

- ▶ Item j can be taken in at most b_j copies
- ▶ A solution is a subset of the extended set of items $\hat{\mathcal{I}}$, with $|\hat{\mathcal{I}}| = \sum_{j \in \mathcal{I}} b_j$
- ▶ Assumptions: not all the copies fit, each item fits alone, all the copies of one item fit

In the example:

- ▶ 8 copies; an optimal solution takes the two copies of item 1 and the copy of item 5, with profit 18

The unbounded knapsack problem

Introduction: the unbounded knapsack problem

- ▶ In the unbounded knapsack problem (UKP) each item is available in an unlimited number of copies
- ▶ The capacity limits the copies anyway: we compute the **maximum feasible number of copies** of each item
- ▶ With it we build, as in the BKP, an extended set of items
- ▶ A solution is again a subset of the extended set

Definition 4: Unbounded Knapsack Problem

Given:

1. a knapsack of capacity $c \in \mathbb{Z}_{\geq 1}$,
2. a set $\mathcal{I} = \{1, 2, \dots, n\}$ of n items,
3. for each item $j \in \mathcal{I}$, which is available in an unlimited number of copies, a profit $p_j \in \mathbb{Z}_{\geq 1}$ and a weight $w_j \in \mathbb{Z}_{\geq 1}$,

the **Unbounded Knapsack Problem (UKP)** asks to decide how many copies of each item to insert into the knapsack, so that the total weight of all selected item copies does not exceed the knapsack capacity and the total profit of all selected item copies is maximized.

UKP instance

A **UKP instance** is specified by:

1. two integer numbers:
the number of items $n \in \mathbb{Z}_{\geq 1}$,
the knapsack capacity $c \in \mathbb{Z}_{\geq 1}$,
2. two vectors of integer numbers of length n :
 $\mathbf{p} \in \mathbb{Z}_{\geq 1}^n$, containing the item profits,
 $\mathbf{w} \in \mathbb{Z}_{\geq 1}^n$, containing the item weights.

Maximum feasible number of copies and extended set

- ▶ Given a UKP instance, we first define for each item $j \in \mathcal{I}$ the **maximum feasible number of copies**

$$\hat{b}_j = \left\lfloor \frac{c}{w_j} \right\rfloor,$$

that is, the largest number of copies of item j that can be inserted into the knapsack without exceeding its capacity.

- ▶ We then define the **extended set of items** as

$$\hat{\mathcal{I}} = \{ (j, k) : j \in \mathcal{I}, k \in \{1, 2, \dots, \hat{b}_j\} \},$$

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

UKP feasible solution

- ▶ The cardinality of the extended set of items is

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

- ▶ A **UKP feasible solution** is a subset of items

$$S \subseteq \hat{\mathcal{J}}$$

such that the total weight of the selected items does not exceed the knapsack capacity, namely

$$\sum_{(j,k) \in S} w_j \leq c.$$

- ▶ Accordingly, the UKP can be equivalently stated as the problem of selecting a subset of the extended set of items whose total weight respects the knapsack capacity, and whose total profit is maximized.

The UKP in formal terms

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

$$\tilde{S} \in \arg \max_{S \subseteq \hat{\mathcal{I}}} \left\{ \sum_{(j,k) \in S} p_j : \sum_{(j,k) \in S} w_j \leq c \right\}.$$

Example (1/4)

We consider a knapsack of capacity $c = 9$ and a set $\mathcal{I} = \{1, 2, 3, 4, 5, 6\}$ of 6 items ($n = 6$), with the following profits and weights:

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
Profits	p_j	6	5	8	9	6	7
Weights	w_j	2	3	6	7	5	9

The maximum feasible number of copies of each item are:

$$\begin{aligned}\hat{b}_1 &= \underbrace{\left\lfloor \frac{9}{2} \right\rfloor}_{=4}, & \hat{b}_2 &= \underbrace{\left\lfloor \frac{9}{3} \right\rfloor}_{=3}, & \hat{b}_3 &= \underbrace{\left\lfloor \frac{9}{6} \right\rfloor}_{=1}, \\ \hat{b}_4 &= \underbrace{\left\lfloor \frac{9}{7} \right\rfloor}_{=1}, & \hat{b}_5 &= \underbrace{\left\lfloor \frac{9}{5} \right\rfloor}_{=1}, & \hat{b}_6 &= \underbrace{\left\lfloor \frac{9}{9} \right\rfloor}_{=1}.\end{aligned}$$

Example (2/4)

The extended set of items is:

$$\hat{\mathcal{I}} = \{ (1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), \\ (3, 1), (4, 1), (5, 1), (6, 1) \}.$$

Each copy (j, k) has profit p_j and weight w_j .

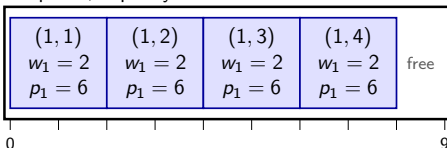
Copy label	1	2	3	4	5	6
(j, k)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(2, 1)	(2, 2)
Profit p_j	6	6	6	6	5	5
Weight w_j	2	2	2	2	3	3
Copy label	7	8	9	10	11	
(j, k)	(2, 3)	(3, 1)	(4, 1)	(5, 1)	(6, 1)	
Profit p_j	5	8	9	6	7	
Weight w_j	3	6	7	5	9	

Example (3/4)

The drawing on the next slide shows an optimal solution:
four copies of item 1, with weight $4 \cdot 2 = 8$ and profit
 $4 \cdot 6 = 24$;
one unit of capacity stays free.

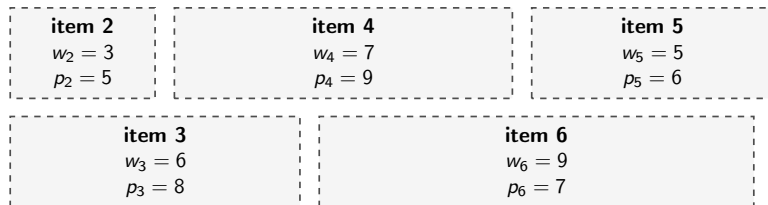
Example (4/4)

knapsack, capacity $c = 9$



weight $4 \cdot 2 = 8$
profit $4 \cdot 6 = 24$

items with no copy selected



Assumptions

We make the following assumptions:

1. The total weight of all possible copies of all items (up to the maximum number that can fit in the knapsack) exceeds the knapsack capacity (otherwise, all copies could be selected).
Formally:

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

2. Each item individually fits into the knapsack (otherwise, it cannot be selected). Formally:

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

Summary

The unbounded knapsack problem:

- ▶ Each item is available in an unlimited number of copies
- ▶ The capacity limits them: at most $\hat{b}_j = \lfloor c/w_j \rfloor$ copies of item j
- ▶ A solution is a subset of the extended set of items, with $|\hat{\mathcal{J}}| = \sum_{j \in \mathcal{J}} \hat{b}_j$

In the example:

- ▶ 11 copies; an optimal solution takes four copies of item 1, with profit 24

The multidimensional knapsack problem

Introduction: the multidimensional knapsack problem

- ▶ In the multidimensional knapsack problem (MDKP) the knapsack has k dimensions, each with its own capacity
- ▶ Each item has a weight in every dimension
- ▶ A subset of items is feasible only if it respects *all* the capacities
- ▶ Several resources consumed together: a common situation in practice

Definition 5: Multidimensional Knapsack Problem

Given:

1. a k -dimensional knapsack,
2. for each dimension $d \in \underbrace{\{1, 2, \dots, k\}}_{=\mathcal{D}}$, a capacity $c_d \in \mathbb{Z}_{\geq 1}$,
3. a set $\mathcal{J} = \{1, 2, \dots, n\}$ of n items,
4. for each item $j \in \mathcal{J}$, a profit $p_j \in \mathbb{Z}_{\geq 1}$,
5. for each dimension $d \in \mathcal{D}$ and each item $j \in \mathcal{J}$, a weight $w_{dj} \in \mathbb{Z}_{\geq 1}$,

the **Multidimensional Knapsack Problem** (MDKP) asks to decide which items to insert into the knapsack, so that for every dimension the total weight of the selected items does not exceed the corresponding capacity, and the total profit of the selected items is maximized.

MDKP instance

An **MDKP instance** is specified by:

1. two integer numbers:
the number of items $n \in \mathbb{Z}_{\geq 1}$,
the number of dimensions $k \in \mathbb{Z}_{\geq 1}$,
2. one vector of integer numbers of length n :
 $\mathbf{p} \in \mathbb{Z}_{\geq 1}^n$, containing the item profits,
3. one vector of integer numbers of length k :
 $\mathbf{c} \in \mathbb{Z}_{\geq 1}^k$, containing the knapsack capacities for each dimension,
4. one matrix of integer numbers with k rows and n columns:
 $\mathbf{W} \in \mathbb{Z}_{\geq 1}^{k \times n}$, containing the weights of the items for each dimension.

MDKP feasible solution

An **MDKP feasible solution** is a subset of items

$$S \subseteq \mathcal{I}$$

such that the capacity constraints are satisfied in every dimension, namely

$$\sum_{j \in S} w_{dj} \leq c_d \quad \forall d \in \mathcal{D}.$$

Accordingly, the MDKP can be equivalently stated as the problem of selecting a subset of items whose total weight respects the knapsack capacities in every dimension, and whose total profit is maximized.

The MDKP in formal terms

Given an MDKP instance, the MDKP can be formally stated as:

$$\tilde{S} \in \arg \max_{S \subseteq \mathcal{I}} \left\{ \sum_{j \in S} p_j \quad : \quad \sum_{j \in S} w_{dj} \leq c_d \quad \forall d \in \mathcal{D} \right\}.$$

Example (1/2)

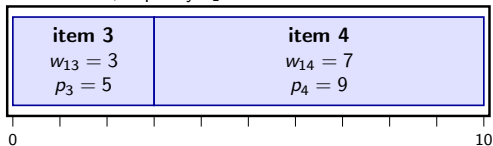
We consider a 2-dimensional knapsack ($k = 2$) with capacities $c_1 = 10$ and $c_2 = 12$, and a set $\mathcal{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following profits and weights:

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$
Profits	p_j	6	7	5	9
Weights (dim. 1)	w_{1j}	4	5	3	7
Weights (dim. 2)	w_{2j}	6	4	5	3

- ▶ The drawing has one box per dimension, of length $c_1 = 10$ and $c_2 = 12$: a selected item occupies a block of length w_{1j} in the first box and one of length w_{2j} in the second.
- ▶ The optimal solution $\{3, 4\}$ fills the first dimension and uses 8 units of the second, with profit $5 + 9 = 14$.

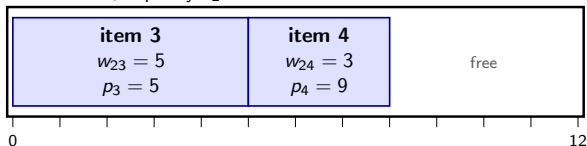
Example (2/2)

dimension 1, capacity $c_1 = 10$



weight $3 + 7 = 10$

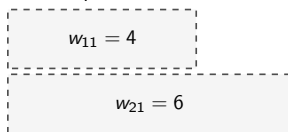
dimension 2, capacity $c_2 = 12$



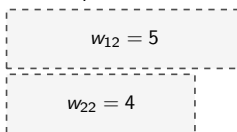
weight $5 + 3 = 8$

not selected (top: dimension 1, bottom: dimension 2)

item 1, $p_1 = 6$



item 2, $p_2 = 7$



Assumptions

We make the following assumptions:

1. The set of all items does *not* fit in the knapsack (otherwise, selecting all items is feasible and optimal). Formally,

$$\exists d \in \mathcal{D} \text{ such that } \sum_{j \in \mathcal{I}} w_{dj} > c_d.$$

2. Each item individually fits in the knapsack in every dimension (otherwise, it can never be selected). Formally,

$$w_{dj} \leq c_d, \quad \forall d \in \mathcal{D}, \forall j \in \mathcal{I}.$$

Summary

The multidimensional knapsack problem:

- ▶ k dimensions with capacities c_d , and weights w_{dj} of item j in dimension d
- ▶ A solution is a subset $S \subseteq \mathcal{I}$ that respects every capacity
- ▶ Assumptions: the set of all items exceeds some capacity, each item fits alone in every dimension

In the example:

- ▶ The optimal solution $\{3, 4\}$ has profit 14

The multiple knapsack problem

Introduction: the multiple knapsack problem

- ▶ In the multiple knapsack problem (MKP) there are m knapsacks, each with its own capacity
- ▶ We decide which items to insert into each knapsack; every item goes into at most one knapsack
- ▶ A solution is a family of pairwise disjoint subsets of items, one for each knapsack

Definition 6: Multiple Knapsack Problem

Given:

1. $m \in \mathbb{Z}_{\geq 1}$ knapsacks,
2. for each knapsack $k \in \underbrace{\{1, 2, \dots, m\}}_{=\mathcal{K}}$, a capacity $c_k \in \mathbb{Z}_{\geq 1}$,
3. a set $\mathcal{J} = \{1, 2, \dots, n\}$ of n items,
4. for each item $j \in \mathcal{J}$, a profit $p_j \in \mathbb{Z}_{\geq 1}$ and a weight $w_j \in \mathbb{Z}_{\geq 1}$,

the **Multiple Knapsack Problem (MKP)** asks to decide which items to insert into each knapsack, so that in every knapsack the total weight of the assigned items does not exceed its capacity, and the total profit of the selected items is maximized.

MKP instance

An **MKP instance** is specified by:

1. two integer numbers:
the number of items $n \in \mathbb{Z}_{\geq 1}$,
the number of knapsacks $m \in \mathbb{Z}_{\geq 1}$,
2. one vector of integer numbers of length n :
 $\mathbf{p} \in \mathbb{Z}_{\geq 1}^n$, containing the item profits,
3. one vector of integer numbers of length n :
 $\mathbf{w} \in \mathbb{Z}_{\geq 1}^n$, containing the item weights,
4. one vector of integer numbers of length m :
 $\mathbf{c} \in \mathbb{Z}_{\geq 1}^m$, containing the capacities of the knapsacks.

MKP feasible solution

An **MKP feasible solution** is a family of subsets of items

$$\mathcal{S} = \{S_k\}_{k \in \mathcal{K}}, \quad S_k \subseteq \mathcal{I} \text{ for all } k \in \mathcal{K},$$

such that:

(i) each item is assigned to at most one knapsack, namely

$$S_k \cap S_{k'} = \emptyset \text{ for all } k, k' \in \mathcal{K} \text{ with } k \neq k';$$

(ii) the total weight of the items assigned to each knapsack respects its capacity, namely

$$\sum_{j \in S_k} w_j \leq c_k \text{ for all } k \in \mathcal{K}.$$

The MKP in formal terms

Accordingly, the MKP can be equivalently stated as the problem of selecting pairwise disjoint subsets of items, one subset for each knapsack, such that every knapsack capacity is respected and the total profit is maximized.

Given an MKP instance, the MKP can be formally stated as:

$$\tilde{\mathcal{S}} \in \arg \max_{\mathcal{S} = \{S_k\}_{k \in \mathcal{K}}} \left\{ \sum_{k \in \mathcal{K}} \sum_{j \in S_k} p_j : \begin{array}{l} S_k \subseteq \mathcal{I} \quad \forall k \in \mathcal{K}, \\ S_k \cap S_{k'} = \emptyset \quad \forall k, k' \in \mathcal{K} \text{ with } k \neq k', \\ \sum_{j \in S_k} w_j \leq c_k \quad \forall k \in \mathcal{K} \end{array} \right\}.$$

Example (1/2)

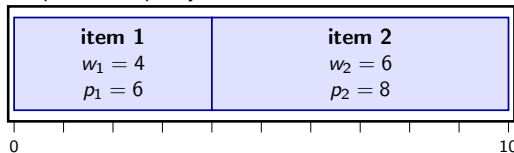
We consider $m = 2$ knapsacks with capacities $c_1 = 10$ and $c_2 = 12$, and a set $\mathcal{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following profits and weights:

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
Profits	p_j	6	8	5	7	9
Weights	w_j	4	6	5	3	7

The drawing on the next slide shows an optimal solution: $S_1 = \{1, 2\}$ in knapsack 1 and $S_2 = \{4, 5\}$ in knapsack 2, with total profit $14 + 16 = 30$; item 3 stays out.

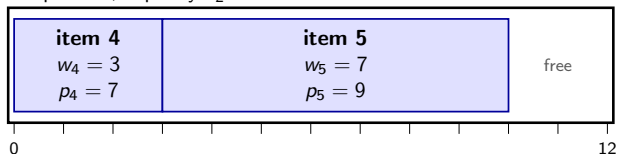
Example (2/2)

knapsack 1, capacity $c_1 = 10$



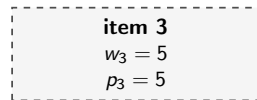
weight $4 + 6 = 10$
profit $6 + 8 = 14$

knapsack 2, capacity $c_2 = 12$



weight $3 + 7 = 10$
profit $7 + 9 = 16$

not selected



Assumptions

We make the following assumptions:

1. The set of all items does *not* fit in the knapsacks altogether (otherwise, selecting all items is feasible and optimal).

Formally,

$$\sum_{j \in \mathcal{I}} w_j > \sum_{k \in \mathcal{K}} c_k.$$

2. Each item individually fits in (at least) one knapsack (otherwise, it can never be selected). Formally,

$$\forall j \in \mathcal{I} \quad \exists k \in \mathcal{K} \text{ such that } w_j \leq c_k.$$

Summary

The multiple knapsack problem:

- ▶ m knapsacks with capacities c_k ; each item goes into at most one knapsack
- ▶ A solution is a family $\{S_k\}_{k \in \mathcal{K}}$ of pairwise disjoint subsets, each within its capacity
- ▶ Assumptions: the items do not fit altogether, each item fits alone in at least one knapsack

In the example:

- ▶ $S_1 = \{1, 2\}$, $S_2 = \{4, 5\}$, total profit 30; item 3 stays out

The multiple-choice knapsack problem

Introduction: the multiple-choice knapsack problem

- ▶ In the multiple-choice knapsack problem (MCKP) the items are partitioned into classes
- ▶ We must select **exactly one** item from each class, within the capacity of the knapsack
- ▶ Because of the word “exactly”, an instance can be infeasible: we see a simple condition that rules this out

Definition 7: Multiple-Choice Knapsack Problem

Given:

1. a knapsack of capacity $c \in \mathbb{Z}_{\geq 1}$,
2. a set $\mathcal{I} = \{1, 2, \dots, n\}$ of n items,
3. the set of items is partitioned into $q \in \mathbb{Z}_{\geq 2}$ classes,
4. for each class $h \in \{1, 2, \dots, q\}$, the subset $\mathcal{I}_h \subseteq \mathcal{I}$ contains the items of that class,
5. for each item $j \in \mathcal{I}$, a profit $p_j \in \mathbb{Z}_{\geq 1}$ and a weight $w_j \in \mathbb{Z}_{\geq 1}$,

the **Multiple-Choice Knapsack Problem** (MCKP) asks to decide which items to insert into the knapsack, so that exactly one item is selected from each class, the total weight of the selected items does not exceed the knapsack capacity, and the total profit of the selected items is maximized.

MCKP instance (1/2)

An **MCKP instance** is specified by:

1. two integer numbers:
the number of items $n \in \mathbb{Z}_{\geq 1}$,
the knapsack capacity $c \in \mathbb{Z}_{\geq 1}$,
2. one vector of integer numbers of length n :
 $\mathbf{p} \in \mathbb{Z}_{\geq 1}^n$, containing the item profits,
3. one vector of integer numbers of length n :
 $\mathbf{w} \in \mathbb{Z}_{\geq 1}^n$, containing the item weights,

MCKP instance (2/2)

4. one integer number:
the number of classes $q \in \mathbb{Z}_{\geq 2}$,
5. one family of subsets:
 $\{\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_q\}$, a partition of $\mathcal{I}$ into q classes, where
 $\mathcal{I}_h \subseteq \mathcal{I}$ is the set of items belonging to class h .

MCKP feasible solution

An **MCKP feasible solution** is a subset of items

$$S \subseteq \mathcal{I}$$

such that:

- (i) exactly one item is selected from each class, namely

$$|S \cap \mathcal{I}_h| = 1 \quad \text{for all } h \in \{1, 2, \dots, q\};$$

- (ii) the total weight of the selected items does not exceed the knapsack capacity, namely

$$\sum_{j \in S} w_j \leq c.$$

The MCKP in formal terms

Accordingly, the MCKP can be equivalently stated as the problem of selecting exactly one item from each class such that the knapsack capacity is respected and the total profit is maximized.

Given an MCKP instance, the MCKP can be formally stated as:

$$\tilde{S} \in \arg \max_{S \subseteq \mathcal{I}} \left\{ \sum_{j \in S} p_j : \sum_{j \in S} w_j \leq c, |S \cap \mathcal{I}_h| = 1 \ \forall h \in \{1, 2, \dots, q\} \right\}.$$

Example (1/2)

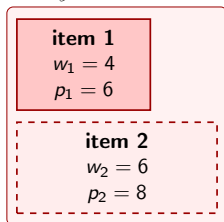
We consider a knapsack of capacity $c = 15$ and a set $\mathcal{I} = \{1, 2, 3, 4, 5, 6\}$ of 6 items ($n = 6$). The items are partitioned into $q = 3$ classes: $\mathcal{I}_1 = \{1, 2\}$, $\mathcal{I}_2 = \{3, 4\}$, $\mathcal{I}_3 = \{5, 6\}$. The profits and weights are given as follows:

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
Profits	p_j	6	8	5	7	4	9
Weights	w_j	4	6	5	8	3	7

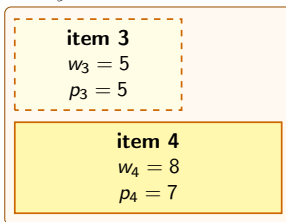
- ▶ In the drawing every class is a coloured group: the item chosen in each class is drawn solid, the other one dashed.
- ▶ An optimal solution, $\{1, 4, 5\}$, fills the knapsack, with weight $4 + 8 + 3 = 15$ and profit $6 + 7 + 4 = 17$.

Example (2/2)

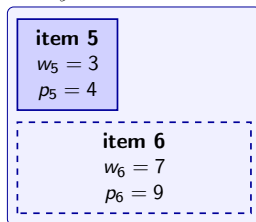
class $\mathcal{I}_1$



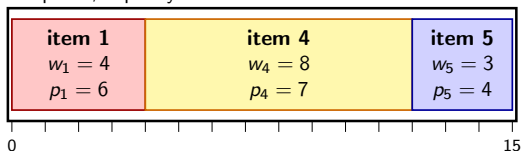
class $\mathcal{I}_2$



class $\mathcal{I}_3$



knapsack, capacity $c = 15$



weight $4 + 8 + 3 = 15$
profit $6 + 7 + 4 = 17$

Assumptions

We make the following assumptions:

1. Each class contains at least one item (otherwise, selecting exactly one item from each class is impossible). Formally:

$$\mathcal{J}_h \neq \emptyset \quad \forall h \in \{1, \dots, q\}.$$

2. The instance is feasible, i.e., there exists at least one choice of exactly one item per class that fits in the knapsack. A necessary and sufficient (and easy-to-check) condition is:

$$\sum_{h=1}^q \min_{j \in \mathcal{J}_h} w_j \leq c.$$

Summary

The multiple-choice knapsack problem:

- ▶ The items are partitioned into q classes $\mathcal{J}_h$
- ▶ A solution selects exactly one item per class, within the capacity
- ▶ Feasibility is not guaranteed: it holds if and only if
$$\sum_{h=1}^q \min_{j \in \mathcal{J}_h} w_j \leq c$$

In the example:

- ▶ Only 3 of the 8 choices fit; $\{1, 4, 5\}$ and $\{2, 3, 5\}$ are optimal, with profit 17

The knapsack problem with setups

Introduction: the knapsack problem with setups

- ▶ In the knapsack problem with setups (KPS) the items are partitioned into classes
- ▶ Using a class has a price: a **setup cost** in the objective and a **setup weight** in the knapsack
- ▶ The setup is paid once for each class with at least one selected item (an **activated class**)
- ▶ The objective is the total revenue minus the setup costs

Definition 8: Knapsack Problem with Setups

Given:

1. a knapsack of capacity $c \in \mathbb{Z}_{\geq 1}$,
2. a set $\mathcal{I} = \{1, 2, \dots, n\}$ of n items,
3. the set of items is partitioned into $q \in \mathbb{Z}_{\geq 2}$ classes,
4. for each class $h \in \{1, 2, \dots, q\}$, the subset $\mathcal{I}_h \subseteq \mathcal{I}$ contains the items of that class, with an associated setup cost $f_h \in \mathbb{Z}_{\geq 1}$ and a setup weight $s_h \in \mathbb{Z}_{\geq 1}$,
5. for each item $j \in \mathcal{I}$, a revenue $p_j \in \mathbb{Z}_{\geq 1}$ and a weight $w_j \in \mathbb{Z}_{\geq 1}$,

the **Knapsack Problem with Setups (KPS)** asks to decide which items to insert into the knapsack, so that the total weight of the selected items plus the setup weights of their classes does not exceed the knapsack capacity, and the total revenue of the selected items minus the setup costs of their classes is maximized.

KPS instance (1/2)

A **KPS instance** is specified by:

1. two integer numbers:
the number of items $n \in \mathbb{Z}_{\geq 1}$,
the knapsack capacity $c \in \mathbb{Z}_{\geq 1}$,
2. one vector of integer numbers of length n :
 $\mathbf{p} \in \mathbb{Z}_{\geq 1}^n$, containing the item revenues,
3. one vector of integer numbers of length n :
 $\mathbf{w} \in \mathbb{Z}_{\geq 1}^n$, containing the item weights,

KPS instance (2/2)

4. one integer number:
the number of classes $q \in \mathbb{Z}_{\geq 2}$,
5. one family of subsets:
 $\{\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_q\}$, a partition of $\mathcal{I}$ into q classes, where
 $\mathcal{I}_h \subseteq \mathcal{I}$ is the set of items belonging to class h ,
6. two vectors of integer numbers of length q :
 $\mathbf{f} \in \mathbb{Z}_{\geq 1}^q$, containing the setup costs of the classes,
 $\mathbf{s} \in \mathbb{Z}_{\geq 1}^q$, containing the setup weights of the classes.

KPS feasible solution

- ▶ A **KPS feasible solution** is a subset of items

$$S \subseteq \mathcal{I}.$$

- ▶ Given S , define the set of **activated classes** as

$$\mathcal{H}(S) := \{ h \in \{1, 2, \dots, q\} : S \cap \mathcal{I}_h \neq \emptyset \}.$$

- ▶ The subset S is feasible if the total weight of the selected items plus the setup weights of the activated classes does not exceed the knapsack capacity, namely

$$\sum_{j \in S} w_j + \sum_{h \in \mathcal{H}(S)} s_h \leq c.$$

The KPS in formal terms

Accordingly, the KPS can be equivalently stated as the problem of selecting a subset of items so as to maximize the total revenue of the selected items minus the setup costs of their activated classes.

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

$$\tilde{S} \in \arg \max_{S \subseteq \mathcal{I}} \left\{ \sum_{j \in S} p_j - \sum_{h \in \mathcal{H}(S)} f_h : \sum_{j \in S} w_j + \sum_{h \in \mathcal{H}(S)} s_h \leq c \right\}.$$

Example (1/3)

- ▶ We consider a knapsack of capacity $c = 20$ and a set $\mathcal{I} = \{1, 2, 3, 4, 5, 6\}$ of 6 items ($n = 6$).
- ▶ The items are partitioned into $q = 2$ classes: $\mathcal{J}_1 = \{1, 2, 3\}$ and $\mathcal{J}_2 = \{4, 5, 6\}$.
- ▶ Each class has a setup cost and a setup weight: $f_1 = 3$, $s_1 = 2$ for class 1, and $f_2 = 4$, $s_2 = 3$ for class 2.
- ▶ The item revenues and weights are given as follows:

Items		$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
Revenues	p_j	6	8	5	7	9	4
Weights	w_j	4	5	3	6	7	2

Example (2/3)

- ▶ In the drawing the classes have two colours, and a darker block is the setup of a class, with its weight s_h and its cost f_h .
- ▶ An optimal solution, $\{1, 2, 3, 6\}$, activates both classes.
- ▶ The weight is

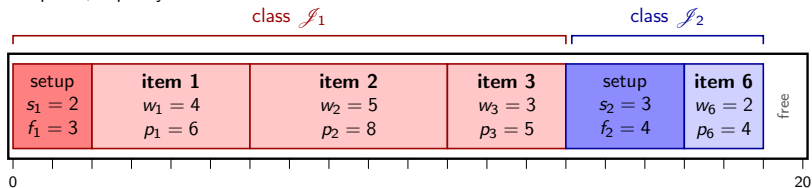
$$(2 + 4 + 5 + 3) + (3 + 2) = 19 \leq 20$$

and the value is

$$(6 + 8 + 5 + 4) - (3 + 4) = 16.$$

Example (3/3)

knapsack, capacity $c = 20$



not selected



Assumptions

We make the following assumptions:

1. Each class contains at least one item. Formally:

$$\mathcal{I}_h \neq \emptyset \quad \forall h \in \{1, 2, \dots, q\}.$$

2. Each item can fit into the knapsack together with the setup weight of its class (otherwise, it can never be selected).

Formally:

$$w_j + s_h \leq c \quad \forall h \in \{1, 2, \dots, q\}, \forall j \in \mathcal{I}_h.$$

Summary

The knapsack problem with setups:

- ▶ Classes $\mathcal{I}_h$ with setup cost f_h and setup weight s_h
- ▶ The setup of a class is paid only if the class is activated, i.e. if at least one of its items is selected
- ▶ Objective: total revenue minus the setup costs of the activated classes

In the example:

- ▶ $\{1, 2, 3\}$, $\{4, 5, 6\}$ and $\{1, 2, 3, 6\}$ are optimal, with value 16

Remarks

Introduction: remarks

- ▶ We close the chapter with three historical remarks on the KP
- ▶ They link this chapter to what comes later in the course: bounds, complexity and approximation

Historical remarks

- ▶ **Dantzig (1957)** sorted the items by non-increasing efficiency p_j/w_j and used this order to solve the continuous version of the KP; the same order returns in the bounds seen later in the course.
- ▶ **Karp (1972)** put the subset sum problem, called knapsack in his paper, among the first 21 NP-complete problems. The KP contains the SSP as the special case $p_j = w_j$, so it is NP-hard.
- ▶ **Ibarra and Kim (1975)** gave the first fully polynomial-time approximation scheme for the KP.

Summary

Remarks:

- ▶ The order of non-increasing efficiency p_j/w_j comes from Dantzig (1957)
- ▶ The KP is NP-hard, since it contains the SSP ($p_j = w_j$), one of Karp's 21 NP-complete problems
- ▶ The KP has a fully polynomial-time approximation scheme (Ibarra and Kim, 1975)

Typical mistakes

Introduction: typical mistakes

- ▶ We list the mistakes that are made most often with knapsack problems
- ▶ They concern the differences among the variants, the order of efficiency and the assumptions on the data
- ▶ Knowing them helps to read and solve the exercises correctly

Observation 2: Typical mistakes (1/3)

- ▶ **Confusing the KP with the SSP.** The SSP asks for a total weight exactly equal to c and has a yes/no answer; it is the KP with $p_j = w_j$, where the answer is yes if the optimal value is c .
- ▶ **Stopping at the order of efficiency.** Taking the items in the order $1, 2, 3, \dots$ of non-increasing p_j/w_j gives $\{1, 2\}$ with profit 11, while the optimal solution of the reference instance is $\{1, 4\}$ with profit 15.

Observation 3: Typical mistakes (2/3)

- ▶ **Counting identical copies as different solutions.** In the BKP and in the UKP the copies (j, k) of the same item are interchangeable: different subsets of the extended set of items can give the same packing.
- ▶ **Choosing at most one item per class.** In the MCKP exactly one item per class is selected, so an instance can be infeasible; in the KPS a class with no selected item costs nothing.

Observation 4: Typical mistakes (3/3)

- ▶ ***Forgetting the assumptions.*** *An item with $w_j > c$ can never be selected, and if all the items fit the problem is trivial: both cases are removed before solving.*

Typical mistakes:

- ▶ The SSP is not the KP: exact target, yes/no answer
- ▶ The order of efficiency does not give an optimal solution
- ▶ Identical copies are interchangeable: count them once
- ▶ MCKP: exactly one item per class; KPS: an unused class costs nothing
- ▶ Remove the trivial cases with the assumptions before solving

In summary

Introduction: in summary

- ▶ We collect the key ideas of the chapter
- ▶ We close with the dashboard of the reference instance: the bounds found so far

Summary (1/2)

- ▶ The KP selects a subset of items of weight at most c and maximum profit; a solution is a subset, and 2^n subsets are candidates.
- ▶ The SSP, the BKP, the UKP, the MDKP, the MKP, the MCKP and the KPS change the number of copies, of resources, of knapsacks or the structure of the items; each one has its assumptions on the data.
- ▶ A solution found by eye is a primal bound; the cardinality bound is a dual bound that uses only the weights and the profits.
- ▶ The order of efficiency is not enough to find an optimal solution, even on a small instance.

Summary (2/2)

Dashboard of the reference instance.

KP, reference instance (max)	lower bound	upper bound	from
this chapter	15 (by eye)	17 (cardinality bound)	definition

- ▶ The absolute gap is 2; listing the 13 feasible subsets shows that $z(I) = 15$.
- ▶ For larger instances, stronger bounds come from relaxations.