

## Proyecto 2: El Problema de la Mochila

Emily Sanchez  
Viviana Vargas

Curso: Investigación de Operaciones  
II Semestre 2025

September 18, 2025

# 1 Problema de la Mochila (Knapsack Problem)

El **problema de la mochila** es un clasico de la *optimizacion combinatoria*. Se dispone de una **mochila** con una **capacidad maxima**  $W$  y un conjunto de  $n$  objetos. Cada objeto  $i$  tiene un **peso**  $w_i$  y un **valor**  $v_i$ . El objetivo es seleccionar los objetos de manera que:

- La suma total de los pesos no exceda la capacidad  $W$ .
- Se maximice el valor total de los objetos elegidos.

## 1.1 Variantes principales

**0/1 Knapsack** Cada objeto puede elegirse una sola vez o no elegirse: decision binaria.

**Bounded Knapsack** Cada objeto puede seleccionarse un numero limitado de veces.

**Unbounded Knapsack** Se permite una cantidad ilimitada de cada objeto.

## 1.2 Solucion

**0/1 Knapsack** Se resuelve comunmente con **programacion dinamica**. Sea  $dp[i][w]$  el valor maximo al considerar los primeros  $i$  objetos y capacidad  $w$ .

$$dp[i][w] = \begin{cases} dp[i-1][w] & \text{si } w_i > w, \\ \max(dp[i-1][w], v_i + dp[i-1][w - w_i]) & \text{si } w_i \leq w. \end{cases}$$

**Unbounded Knapsack** Similar al 0/1 pero permitiendo repeticiones:

$$dp[w] = \max(dp[w], v_i + dp[w - w_i]).$$

**Tipo de problema:** Bounded Knapsack  
**Capacidad máxima:** 13  
**Número de objetos:** 5

## Datos del Problema

| Objeto | Costo | Valor | Cantidad |
|--------|-------|-------|----------|
| A      | 10.00 | 20.00 | 1        |
| B      | 3.00  | 12.00 | 3        |
| C      | 5.00  | 10.00 | 1        |
| D      | 5.00  | 6.00  | 4        |
| E      | 9.00  | 11.00 | 1        |

## Tabla de Programación Dinámica Detallada

| Capacidad | A     | B     | C  | D  | E  |
|-----------|-------|-------|----|----|----|
| 0         | 0     | 0     | 0  | 0  | 0  |
| 1         | 0     | 0     | 0  | 0  | 0  |
| 2         | 0     | 0     | 0  | 0  | 0  |
| 3         | 0     | 12(1) | 12 | 12 | 12 |
| 4         | 0     | 12(1) | 12 | 12 | 12 |
| 5         | 0     | 12(1) | 12 | 12 | 12 |
| 6         | 0     | 24(2) | 24 | 24 | 24 |
| 7         | 0     | 24(2) | 24 | 24 | 24 |
| 8         | 0     | 24(2) | 24 | 24 | 24 |
| 9         | 0     | 36(3) | 36 | 36 | 36 |
| 10        | 20(1) | 36(3) | 36 | 36 | 36 |
| 11        | 20(1) | 36(3) | 36 | 36 | 36 |
| 12        | 20(1) | 36(3) | 36 | 36 | 36 |
| 13        | 20(1) | 36(3) | 36 | 36 | 36 |

## Solución Óptima

**Valor máximo obtenido:** 36  
**Objetos seleccionados:** B:3  
**Capacidad utilizada:** 9