

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 .

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

Bounded Knapsack Similar al 0/1 pero puede tener uno o más cantidades por objeto. Es limitado, por lo que no puede ser infinito.

$$dp[i][w] = \max_{0 \leq k \leq c_i, k w_i \leq w} (dp[i-1][w - k w_i] + k v_i).$$

Unbounded Knapsack Similar al bounded pero permitiendo repeticiones sin limite de cantidades (infinito).

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

Tipo de problema: 0/1 Knapsack
 Capacidad máxima: 12
 Número de objetos: 14

Datos del Problema

Objeto	Costo	Valor	Cantidad
A	4.00	15.00	1
B	12.00	21.00	1
C	10.00	11.00	1
D	9.00	5.00	1
E	3.00	15.00	1
F	2.00	7.00	1
G	11.00	8.00	1
H	8.00	9.00	1
I	7.00	12.00	1
J	3.00	4.00	1
K	4.00	10.00	1
L	5.00	22.00	1
M	9.00	13.00	1
N	1.00	9.00	1

Tabla de Programación Dinámica

Capacidad/Objetos	A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	9
2	0	0	0	0	0	7	7	7	7	7	7	7	7	9
3	0	0	0	0	15	15	15	15	15	15	15	15	15	16
4	15	15	15	15	15	15	15	15	15	15	15	15	15	24
5	15	15	15	15	15	22	22	22	22	22	22	22	22	24
6	15	15	15	15	15	22	22	22	22	22	22	22	22	31
7	15	15	15	15	30	30	30	30	30	30	30	30	30	31
8	15	15	15	15	30	30	30	30	30	30	30	37	37	39
9	15	15	15	15	30	37	37	37	37	37	37	37	37	46
10	15	15	15	15	30	37	37	37	37	37	37	44	44	46
11	15	15	15	15	30	37	37	37	37	37	40	44	44	53
12	15	21	21	21	30	37	37	37	37	41	41	52	52	53

Solución Óptima

Valor máximo obtenido: 53
 Objetos seleccionados: N, L, F, A
 Capacidad utilizada: 12