

# Knapsack Problem

PROGETTO  
Laboratorio di Intelligenza Artificiale

Carolina Crespi

January 28, 2026

## 1 Knapsack Problem – KP

Il *Knapsack Problem* (KP) è uno dei più noti problemi di ottimizzazione combinatoria ed è un classico esempio di problema *NP-hard*. Il problema consiste nel selezionare un sottoinsieme di oggetti, ciascuno caratterizzato da un peso e da un valore, al fine di massimizzare il valore totale senza superare una capacità massima disponibile. Formalmente, sia dato un insieme di  $n$  oggetti, dove a ciascun oggetto  $i$  sono associati:

- un peso  $w_i > 0$ ,
- un valore  $v_i > 0$ .

Sia inoltre dato un valore  $W$  che rappresenta la capacità massima dello zaino. Il problema consiste nel determinare una soluzione binaria

$$x = (x_1, x_2, \dots, x_n), \quad x_i \in \{0, 1\},$$

tale da massimizzare la funzione obiettivo:

$$\max \sum_{i=1}^n v_i x_i$$

soggetta al vincolo:

$$\sum_{i=1}^n w_i x_i \leq W.$$

Nel caso classico *0/1 Knapsack*, ogni oggetto può essere selezionato al più una volta.

## 2 Istanza del Problema

Nel progetto viene considerata una singola istanza del problema 0/1 Knapsack, fornita insieme alla traccia. L'istanza è composta da cinque oggetti, ciascuno con peso e valore assegnati, e da una capacità massima dello zaino  $W$ . Per l'istanza considerata è noto il valore ottimo della funzione obiettivo, che può essere utilizzato per verificare la correttezza delle soluzioni ottenute.

## 3 Protocollo Sperimentale

Per l'istanza considerata, l'algoritmo scelto deve essere eseguito utilizzando come criterio di arresto:

- il raggiungimento del valore ottimo noto della funzione obiettivo, **oppure**

- il raggiungimento di un numero massimo di 200 valutazioni della funzione obiettivo.

Per gli algoritmi deterministici (Branch and Bound) è richiesta una sola esecuzione dell'algoritmo.

Per gli algoritmi stocastici (Local Search, Algoritmo Genetico e Simulated Annealing), l'algoritmo deve essere eseguito per **5 run** indipendenti, utilizzando condizioni iniziali o semi casuali diversi.

Nella relazione finale devono essere riportati:

- il valore della funzione obiettivo ottenuto;
- la soluzione individuata (vettore binario);
- la verifica del rispetto del vincolo di capacità;
- per gli algoritmi stocastici, la media dei valori finali ottenuti sulle 5 run;
- un breve commento sul comportamento dell'algoritmo adottato.

## 4 Algoritmo da Implementare

Il candidato è invitato a scegliere e implementare **uno** dei seguenti approcci per la risoluzione del Knapsack Problem:

1. **Branch and Bound**, con definizione dell'albero delle decisioni e strategie di esplorazione dello spazio di ricerca (depth-first, best-bound-first o strategia mista);
2. **Local Search**, basata su un vicinato definito dal flip di un bit e sull'accettazione di sole soluzioni migliorative;
3. **Algoritmo Genetico (GA)** con codifica binaria della soluzione, selezione (roulette wheel o torneo di dimensione 2), crossover a un punto e mutazione mediante flip di un bit;
4. **Simulated Annealing (SA)** con rappresentazione binaria della soluzione, vicinato basato sul flip di un bit, temperatura iniziale elevata, condizione di equilibrio con 5 mosse per ciascun valore di temperatura, raffreddamento lineare.

La scelta dell'algoritmo deve essere motivata nella relazione finale, insieme alla descrizione delle principali scelte progettuali effettuate.

## 5 Note Finali al Progetto

La relazione finale deve essere redatta in L<sup>A</sup>T<sub>E</sub>X, con una lunghezza minima di **4 pagine** e massima di **12 pagine**. È fortemente sconsigliata l'inclusione di parti di codice implementate, mentre è fortemente consigliato l'uso di pseudo-codice e diagrammi esplicativi.

La relazione deve illustrare in modo chiaro e dettagliato:

- l'algoritmo sviluppato e le sue caratteristiche principali;
- le motivazioni delle scelte progettuali effettuate;
- eventuali elementi di novità e originalità introdotti;
- un'analisi critica dei risultati ottenuti sull'istanza assegnata;
- considerazioni e conclusioni personali sul progetto sviluppato.

La valutazione finale si baserà principalmente su:

- originalità e complessità dell'algoritmo sviluppato;

- qualità dei risultati ottenuti;
- qualità espositiva della relazione;
- completezza e chiarezza dei contenuti.

**Consegna:** il progetto (codice sorgente) e la relazione finale (.pdf e .tex, inclusi eventuali file di figure e/o tabelle) devono essere inviati via email all'indirizzo `carolina.crespi@unict.it` entro le ore **14:00 del 16 febbraio 2026**.

Il candidato è invitato a comunicare la scelta dell'algoritmo da sviluppare entro le ore **16:00 del 30 gennaio 2026**.