

# Bin Packing Problem

## PROGETTO

Laboratorio di Intelligenza Artificiale

Carolina Crespi

24 Giugno 2026

## 1 Bin Packing Problem – BPP

Il *Bin Packing Problem* (BPP) è un classico problema di ottimizzazione combinatoria, classificato come NP-hard, con numerose applicazioni in informatica e ricerca operativa (allocazione di memoria, scheduling, taglio di materiali, logistica).

Formalmente, sia dato un insieme di  $n$  oggetti, dove a ciascun oggetto  $i$  è associato un peso  $w_i > 0$ , e una capacità  $C > 0$  comune a un numero illimitato di *bin* (contenitori) identici. Il problema consiste nel determinare il minimo numero di bin necessari per allocare tutti gli oggetti, in modo tale che la somma dei pesi degli oggetti assegnati a ciascun bin non superi la capacità  $C$ .

Indicando con  $K$  il numero di bin utilizzati e con  $B_k$  il sottoinsieme di oggetti assegnati al  $k$ -esimo bin, la funzione obiettivo è:

$$\min K \quad (1)$$

soggetta ai vincoli:

$$\sum_{i \in B_k} w_i \leq C \quad \forall k \in \{1, \dots, K\}, \quad (2)$$

$$\bigcup_{k=1}^K B_k = \{1, \dots, n\}, \quad (3)$$

$$B_k \cap B_h = \emptyset \quad \forall k \neq h. \quad (4)$$

Ogni oggetto deve dunque essere assegnato a esattamente un bin, e ogni bin deve rispettare il vincolo di capacità.

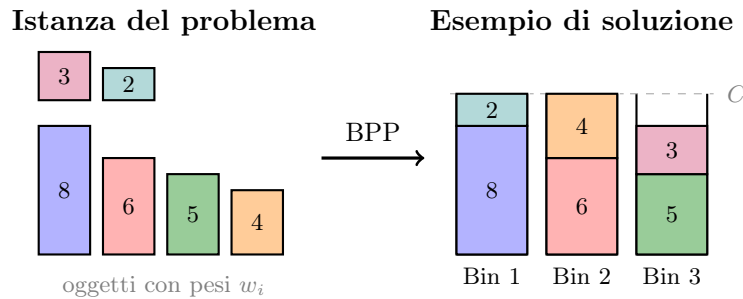


Figura 1: A sinistra: un insieme illustrativo di oggetti con peso variabile. A destra: un esempio di soluzione che alloca gli oggetti in bin di capacità  $C$  (linea tratteggiata) rispettando il vincolo di capacità. La figura ha scopo unicamente illustrativo e non corrisponde all'istanza considerata nel progetto.

## 2 Istanza del Problema

Nel progetto è considerata una singola istanza del BPP composta da  $n = 12$  oggetti e capacità  $C = 15$  per ciascun bin.

I dati dell'istanza sono forniti nel file allegato alla presente traccia:

- **items.txt**: contiene il numero di oggetti  $n$ , la capacità  $C$  e, per ciascun oggetto  $i \in \{1, \dots, n\}$ , il valore del peso  $w_i$ .

Il valore ottimo della funzione obiettivo è noto ed è pari a  $K^* = 5$  bin. Tale valore può essere utilizzato per verificare la correttezza delle soluzioni ottenute.

## 3 Protocollo Sperimentale

Il protocollo sperimentale varia in funzione della natura dell'algoritmo scelto.

### Algoritmi deterministici (Branch and Bound, Euristica Costruttiva)

Per gli algoritmi deterministici è richiesta **una sola esecuzione**. Nella relazione finale devono essere riportati:

- il numero di bin utilizzati nella soluzione trovata;
- l'assegnazione esplicita degli oggetti ai bin;
- la verifica del rispetto del vincolo di capacità per ciascun bin.

Per il Branch and Bound devono essere inoltre descritte la strategia di esplorazione adottata e il criterio di *lower bound* utilizzato.

### Algoritmi stocastici (Simulated Annealing, Algoritmo Genetico)

Per gli algoritmi stocastici sono richieste **5 run indipendenti**, utilizzando seed diversi. Ogni run è governata dal seguente criterio di arresto:

- il raggiungimento del valore ottimo noto  $K^* = 5$ , oppure
- il raggiungimento di un numero massimo di **500 valutazioni** della funzione obiettivo.

Una *valutazione della funzione obiettivo* corrisponde al conteggio del numero di bin utilizzati da una soluzione completa e ammissibile (assegnazione di tutti gli oggetti che rispetta il vincolo di capacità).

Nella relazione finale devono essere riportati:

- il numero di bin ottenuto in ciascuna run;
- l'assegnazione degli oggetti ai bin della run migliore;
- la verifica del rispetto del vincolo di capacità per ciascun bin;
- la media dei valori finali 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 algoritmi:

1. **Branch and Bound (BnB)**, con definizione dell'albero delle decisioni e strategia di esplorazione dello spazio di ricerca (*depth-first*, *best-bound-first* o strategia mista). La strategia e il criterio di *lower bound* adottati devono essere descritti nella relazione.
2. **Euristica Costruttiva Greedy**, in cui gli oggetti vengono processati in sequenza (eventualmente dopo un ordinamento per peso) e ciascuno è assegnato a un bin esistente secondo un criterio fissato: ad esempio, il primo bin in cui l'oggetto entra, oppure il bin con il minor spazio residuo che lo possa contenere. Se nessun bin disponibile può contenere l'oggetto, ne viene aperto uno nuovo. La strategia di selezione del bin e l'eventuale criterio di ordinamento devono essere dichiarati e motivati nella relazione.
3. **Simulated Annealing (SA)**, con rappresentazione della soluzione come vettore di assegnazione oggetto-bin, vicinato basato sullo spostamento di un oggetto da un bin a un altro (o sullo scambio tra due oggetti in bin distinti), temperatura iniziale, schema di raffreddamento e condizione di equilibrio lasciati alla scelta del candidato. È obbligatorio garantire il rispetto del vincolo di capacità in ogni soluzione generata. Tutti i parametri devono essere dichiarati e motivati nella relazione.
4. **Algoritmo Genetico (GA)**, con codifica della soluzione, operatori di selezione, crossover e mutazione lasciati alla scelta del candidato. È obbligatorio garantire il rispetto del vincolo di capacità in ogni soluzione generata. Tutti i parametri devono essere dichiarati e motivati nella relazione.

## 5 Letture Consigliate

Per approfondire il problema e gli algoritmi proposti, si suggeriscono i seguenti riferimenti:

- Martello, S., & Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. Wiley. (Cap. 8 dedicato al Bin Packing.)
- Coffman, E. G., Csirik, J., Galambos, G., Martello, S., & Vigo, D. (2013). Bin packing approximation algorithms: survey and classification. In *Handbook of Combinatorial Optimization*, Springer.
- Falkenauer, E. (1996). A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2(1), 5–30.
- Korte, B., & Vygen, J. (2018). *Combinatorial Optimization: Theory and Algorithms* (6th ed.). Springer.

## 6 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 codice implementato, mentre è fortemente consigliato l'uso di pseudo-codice e diagrammi esplicativi.

La relazione deve illustrare in modo chiaro:

- 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;
- considerazioni finali sul comportamento dell'algoritmo.

La valutazione finale si baserà principalmente su:

- correttezza dell'implementazione e rispetto dei vincoli;
- qualità dei risultati ottenuti;
- qualità espositiva della relazione;
- completezza e chiarezza dell'analisi.

**Consegna:** il progetto (codice sorgente) e la relazione finale (**.pdf** e **.tex**) devono essere inviati via email all'indirizzo **carolina.crespi@unict.it** entro le ore 14:00 del **13 luglio 2026**. Il candidato è invitato a comunicare la scelta dell'algoritmo da sviluppare entro le ore 16:00 del **25 giugno 2026**.