

Multi-dimensional Knapsack Problem

PROGETTO

Laboratorio di Intelligenza Artificiale

Carolina Crespi

15 Luglio 2026

1 Multi-dimensional Knapsack Problem – MKP

Il *Multi-dimensional Knapsack Problem* (MKP) è una generalizzazione del classico Knapsack Problem, in cui la selezione degli oggetti è soggetta a più vincoli di capacità simultanei. Come il KP monodimensionale, è un problema di ottimizzazione combinatoria NP-hard.

Intuitivamente, il MKP può essere pensato come il problema di riempire uno zaino nel quale gli oggetti devono rispettare contemporaneamente più limiti: ad esempio, un limite di peso e un limite di volume. Questa formulazione modella in modo naturale molti scenari applicativi, quali:

- il carico di un veicolo, soggetto sia a una portata massima sia a uno spazio limitato;
- l'allocazione di un budget su un insieme di progetti che consumano risorse eterogenee (denaro, tempo, personale);
- la selezione di un portafoglio di investimenti in presenza di più indicatori di rischio o di esposizione;
- l'assegnazione di task a un sistema di calcolo con vincoli congiunti su memoria, CPU e banda.

Formalmente, sia dato un insieme di n oggetti, dove a ciascun oggetto i sono associati:

- un valore $v_i > 0$;
- m pesi $w_{i,1}, w_{i,2}, \dots, w_{i,m} > 0$, uno per ciascuna dimensione (ad esempio peso, volume, costo).

Siano inoltre dati m valori di capacità $W_1, W_2, \dots, W_m$, uno per ciascuna dimensione.

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:

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

soggetta agli m vincoli di capacità:

$$\sum_{i=1}^n w_{i,j} x_i \leq W_j \quad \forall j \in \{1, \dots, m\}. \tag{2}$$

Ogni oggetto può dunque essere selezionato al più una volta, e la selezione deve rispettare simultaneamente tutti i vincoli di capacità.

Rispetto al KP classico, la presenza di più dimensioni introduce una difficoltà strutturale non banale. Mentre nel KP monodimensionale la sola densità v_i/w_i fornisce un criterio guida efficace per la selezione degli oggetti, nel MKP un oggetto con alta densità rispetto a una dimensione può risultare svantaggioso rispetto a un'altra: la scelta ottima nasce da un compromesso tra i diversi vincoli.

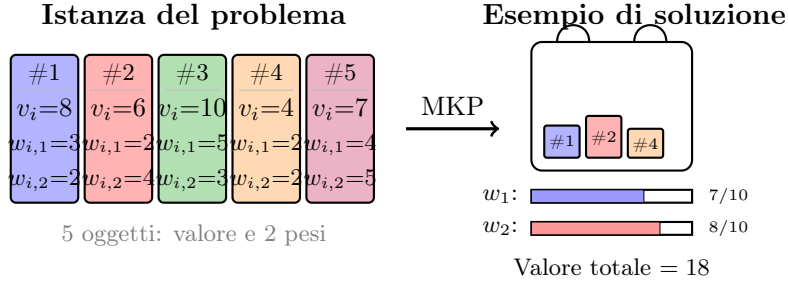


Figura 1: A sinistra: un insieme illustrativo di oggetti, ciascuno caratterizzato da un valore v_i e da due pesi $w_{i,1}, w_{i,2}$ (ad esempio peso e volume). A destra: un esempio di soluzione ammissibile che seleziona alcuni oggetti, con l'utilizzo delle due capacità W_1 e W_2 rappresentato dalle barre orizzontali. 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 MKP composta da $n = 10$ oggetti e $m = 2$ dimensioni (peso e volume), con capacità $W_1 = 18$ e $W_2 = 18$.

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

- **items.txt**: contiene il numero di oggetti n , il numero di dimensioni m , le capacità W_1, W_2 e, per ciascun oggetto $i \in \{1, \dots, n\}$, il valore v_i e i pesi $w_{i,1}, w_{i,2}$.

Il valore ottimo della funzione obiettivo è noto ed è pari a $v^* = 46$. 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 valore della funzione obiettivo ottenuto;
- la soluzione individuata (vettore binario);
- la verifica del rispetto di entrambi i vincoli di capacità.

Per il Branch and Bound devono essere inoltre descritte la strategia di esplorazione adottata e il criterio di *upper 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 $v^* = 46$, oppure
- il raggiungimento di un numero massimo di **300 valutazioni** della funzione obiettivo.

Una *valutazione della funzione obiettivo* corrisponde al calcolo del valore totale di una soluzione completa e ammissibile (che rispetta entrambi i vincoli di capacità).

Nella relazione finale devono essere riportati:

- il valore della funzione obiettivo ottenuto in ciascuna run;
- la soluzione della run migliore (vettore binario);
- la verifica del rispetto di entrambi i vincoli;
- 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 (una variabile x_i per livello) e strategia di esplorazione dello spazio di ricerca (*depth-first*, *best-bound-first* o strategia mista). La strategia e il criterio di *upper bound* adottati devono essere descritti nella relazione.
2. **Euristica Costruttiva Greedy**, in cui gli oggetti vengono ordinati secondo un criterio di priorità definito dal candidato (ad esempio, in base al rapporto tra valore e una misura aggregata dei pesi lungo le m dimensioni) e successivamente aggiunti in sequenza alla soluzione, purché il loro inserimento rispetti entrambi i vincoli di capacità. Il criterio di ordinamento adottato deve essere dichiarato e motivato nella relazione.
3. **Simulated Annealing (SA)**, con rappresentazione binaria della soluzione, vicinato basato sul flip di un bit, temperatura iniziale, schema di raffreddamento e condizione di equilibrio lasciati alla scelta del candidato. È obbligatorio garantire il rispetto di entrambi i vincoli di capacità in ogni soluzione generata (ad esempio, mediante procedura di riparazione o penalizzazione). Tutti i parametri devono essere dichiarati e motivati nella relazione.
4. **Algoritmo Genetico (GA)**, con codifica binaria della soluzione, operatori di selezione, crossover e mutazione lasciati alla scelta del candidato. È obbligatorio garantire il rispetto di entrambi i vincoli di capacità in ogni soluzione generata. Tutti i parametri devono essere dichiarati e motivati nella relazione.

5 Note Finali al Progetto

La relazione finale deve essere redatta in $\text{\LaTeX}$, 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 **31 luglio 2026**. Il candidato è invitato a comunicare la scelta dell'algoritmo da sviluppare entro le ore 16:00 del **18 luglio 2026**.