

Capacitated Vehicle Routing Problem

PROGETTO

Laboratorio di Intelligenza Artificiale

Carolina Crespi

30 Marzo 2026

1 Capacitated Vehicle Routing Problem – CVRP

Il **Capacitated Vehicle Routing Problem** (CVRP) è un classico problema di ottimizzazione combinatoria, classificato come NP-hard, che generalizza il Traveling Salesman Problem introducendo una flotta di veicoli con capacità limitata.

Formalmente, si consideri un grafo non orientato $G = (V, E)$, dove:

- $V = \{0, 1, \dots, n\}$ è l'insieme dei nodi: il nodo 0 rappresenta il **deposito** (depot), mentre i nodi $1, \dots, n$ rappresentano i **clienti**;
- E è l'insieme degli archi, a ciascuno dei quali è associata una distanza $d_{ij} > 0$, con $d_{ij} = d_{ji}$ (caso simmetrico).

Ogni cliente $i \in \{1, \dots, n\}$ è associato a una **domanda** $q_i > 0$ che deve essere soddisfatta. È disponibile una flotta di K veicoli identici, ciascuno con capacità massima Q . Ogni veicolo parte dal deposito, visita un sottoinsieme di clienti e ritorna al deposito, senza mai superare la capacità Q .

L'obiettivo è determinare un insieme di al più K rotte che:

- partano e terminino al deposito (nodo 0);
- visitino ciascun cliente esattamente una volta;
- rispettino il vincolo di capacità per ogni veicolo;
- minimizzino la **distanza totale** percorsa da tutti i veicoli.

Indicando con R_k la rotta del k -esimo veicolo, la funzione obiettivo è:

$$\min \sum_{k=1}^K \sum_{\text{archi } (i,j) \in R_k} d_{ij} \quad (1)$$

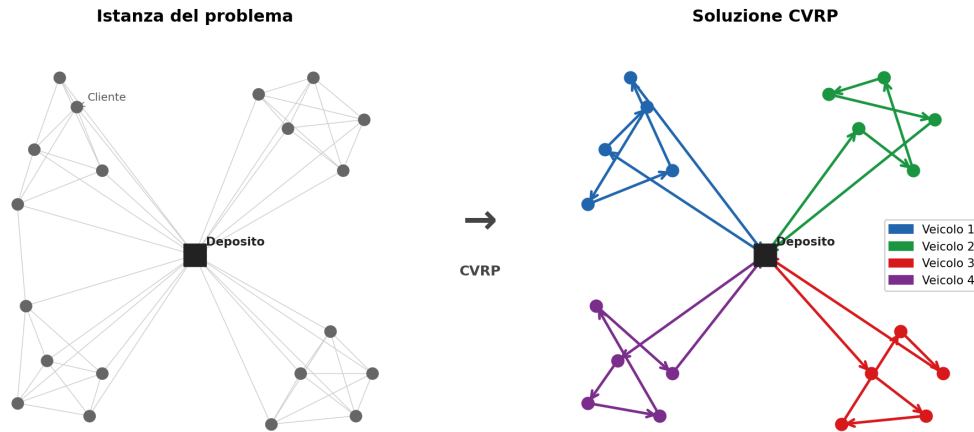


Figura 1: A sinistra: istanza del CVRP con deposito e clienti collegati da un grafo completo. A destra: esempio di soluzione con tre rotte distinte, ciascuna con partenza e ritorno al deposito.

2 Istanza del Problema

Nel progetto è considerata una singola istanza del CVRP simmetrico con $n = 8$ clienti e un deposito (nodo 0), per un totale di 9 nodi.

I dati dell'istanza sono forniti in due file allegati alla presente traccia:

- **distances.txt**: matrice delle distanze $D = (d_{ij})$, di dimensione 9×9 , simmetrica con diagonale nulla;
- **customers.txt**: tabella con, per ciascun cliente (1–8), il valore della domanda q_i .

I parametri della flotta sono:

- numero massimo di veicoli disponibili: $K = 3$;
- capacità massima per veicolo: $Q = 20$.

Il valore ottimo della funzione obiettivo è noto ed è pari a **82.8**. 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, Nearest Neighbor)

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

- il valore della funzione obiettivo ottenuto;
- le rotte individuate, con indicazione esplicita della sequenza dei clienti visitati da ciascun veicolo, inclusa la partenza e il ritorno al deposito;

- la verifica del rispetto del vincolo di capacità per ogni rotta.

Per il **Branch and Bound** deve essere inoltre descritta la strategia di esplorazione adottata e il criterio di bounding utilizzato.

Algoritmi stocastici (Local Search, 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 della funzione obiettivo (82.8), oppure
- il raggiungimento di un numero massimo di **500 valutazioni** della funzione obiettivo.

Una *valutazione della funzione obiettivo* corrisponde al calcolo della distanza totale di una soluzione completa e ammissibile (insieme di rotte che copre tutti i clienti nel rispetto del vincolo di capacità).

Nella relazione finale devono essere riportati:

- il valore della funzione obiettivo ottenuto in ciascuna run;
- le rotte individuate dalla run migliore;
- la verifica del rispetto del vincolo di capacità per ogni rotta;
- 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. **Nearest Neighbor Heuristic (NN)**, con costruzione greedy delle rotte a partire dal deposito.
3. **Local Search (LS)**, con soluzione iniziale costruita tramite la Nearest Neighbor Heuristic e vicinato definito a scelta dal candidato tra *2-opt* e *swap*, nel rispetto del vincolo di capacità. È richiesto l'utilizzo del criterio di accettazione *best-improvement* oppure *first-improvement*, da specificare 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 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 **17 aprile 2026**. Il candidato è invitato a comunicare la scelta dell'algoritmo da sviluppare entro le ore 16:00 del **1 aprile 2026**.