

Traveling Salesman Problem

PROGETTO

Laboratorio di Intelligenza Artificiale

Carolina Crespi

February 18, 2026

1 Traveling Salesman Problem – TSP

Il *Traveling Salesman Problem* (TSP) è uno dei più noti problemi di ottimizzazione combinatoria ed è un classico esempio di problema *NP-hard*.

Dato un insieme di n città e una matrice delle distanze $D = (d_{ij})$, dove $d_{ij} > 0$ rappresenta la distanza tra la città i e la città j , il problema consiste nel determinare un percorso chiuso (tour) di lunghezza minima che:

- visiti ciascuna città esattamente una volta;
- ritorni alla città di partenza.

Formalmente, si vuole determinare una permutazione π delle città tale da minimizzare la funzione obiettivo:

$$\min \sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)}.$$

Nel presente progetto si considera il caso di **TSP simmetrico**, ossia $d_{ij} = d_{ji}$ per ogni coppia di città.

2 Istanza del Problema

Nel progetto viene considerata una singola istanza del TSP simmetrico composta da **5 città**, fornita insieme alla traccia. Per l'istanza considerata è noto il valore ottimo della funzione obiettivo, pari a: **19**. Tale valore 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 **100 valutazioni** della funzione obiettivo.

Una **valutazione della funzione obiettivo** corrisponde al calcolo della lunghezza di un tour completo. Per algoritmi deterministici è richiesta una sola esecuzione. Per algoritmi stocastici sono richieste **5 run indipendenti**, utilizzando seed diversi.

Nella relazione finale devono essere riportati:

- il valore della funzione obiettivo ottenuto;
- il tour individuato (permutazione delle città);
- per algoritmi stocastici, 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**, con definizione dell'albero delle decisioni e strategia di esplorazione dello spazio di ricerca (depth-first, best-bound-first o strategia mista). La strategia deve essere descritta nella relazione.
2. **Local Search**, con rappresentazione data dalla permutazione delle città, soluzione iniziale generata casualmente e vicinato definito a scelta dal candidato. È richiesto l'utilizzo del criterio di accettazione **best-improvement** oppure **first-improvement**, da specificare nella relazione.
3. **Simulated Annealing (SA)**, con rappresentazione basata su permutazioni, soluzione iniziale casuale e vicinato dato dallo scambio di due città. I parametri dell'algoritmo sono lasciati alla scelta del candidato e devono essere **esplicitamente dichiarati e motivati** nella relazione.
4. **Ant Colony Optimization (ACO)**, con numero di formiche, parametri di controllo (α , β , ρ), strategia di aggiornamento del feromone e ogni altro parametro rilevante lasciati alla scelta del candidato e da **dichiarare e motivare** nella relazione.

5 Note Finali al Progetto

La relazione finale deve essere redatta in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, 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;
- 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 9 marzo 2026. Il candidato è invitato a comunicare la scelta dell'algoritmo da sviluppare entro le ore 16:00 del 20 febbraio 2026.