

Graph Coloring Problem

PROGETTO

Laboratorio di Intelligenza Artificiale

Carolina Crespi

29 Aprile 2026

1 Graph Coloring Problem – GCP

Il **Graph Coloring Problem** (GCP) è un classico problema di ottimizzazione combinatoria, classificato come NP-hard, con numerose applicazioni in informatica e ricerca operativa.

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

- $V = \{0, 1, \dots, n - 1\}$ è l'insieme dei **nodi**;
- $E \subseteq V \times V$ è l'insieme degli **archi**,

si vuole assegnare a ciascun nodo un **colore** (intero positivo) tale che:

- nodi adiacenti (collegati da un arco) ricevano colori distinti;
- il numero totale di colori utilizzati sia minimizzato.

Il minimo numero di colori necessario per colorare il grafo è detto **numero cromatico** $\chi(G)$. Formalmente, si cerca una funzione $c : V \rightarrow \{1, \dots, k\}$ tale da minimizzare k soggetta al vincolo:

$$c(u) \neq c(v) \quad \forall (u, v) \in E. \quad (1)$$

2 Istanza del Problema

Nel progetto è considerata una singola istanza del GCP composta da $n = 11$ nodi e 23 archi.

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

- **graph.txt**: la prima riga contiene il numero di nodi n ; le righe successive contengono la matrice di adiacenza $A \in \{0, 1\}^{n \times n}$, simmetrica con diagonale nulla, dove $A_{ij} = 1$ indica la presenza dell'arco (i, j) .

Il numero cromatico dell'istanza è noto ed è pari a $\chi(G) = 3$. Tale valore può essere utilizzato per verificare la correttezza delle soluzioni ottenute.

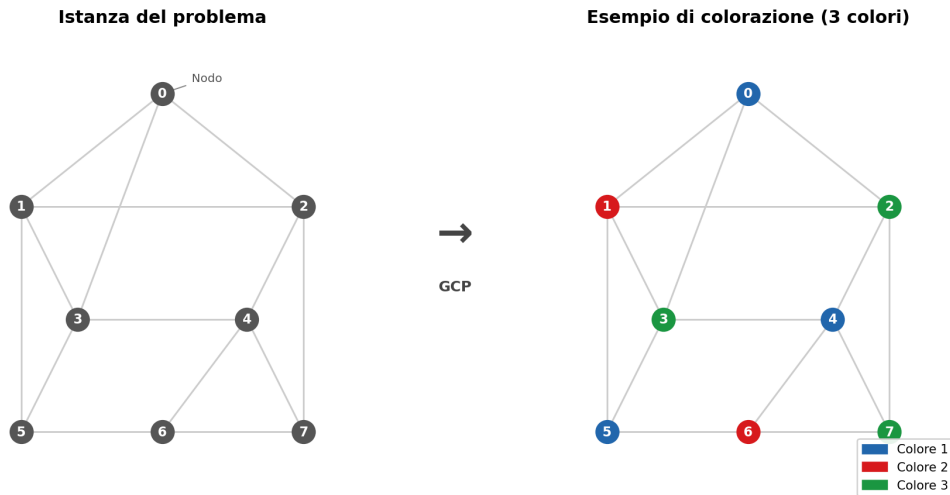


Figura 1: A sinistra: istanza del GCP. A destra: esempio di colorazione ottima con 3 colori. Nodi adiacenti ricevono sempre colori distinti.

3 Protocollo Sperimentale

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

Algoritmo deterministico (Branch and Bound)

Per il Branch and Bound è richiesta **una sola esecuzione**. Nella relazione finale devono essere riportati:

- il numero di colori utilizzati nella soluzione trovata;
- l'assegnazione dei colori ai nodi (vettore c);
- la verifica dell'ammissibilità della soluzione (nessun arco monocromatico);
- la strategia di esplorazione adottata e il criterio di *lower bound* utilizzato.

Algoritmi stocastici (Local Search, 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 numero cromatico noto $\chi(G) = 3$, oppure
- il raggiungimento di un numero massimo di **500 valutazioni** della funzione obiettivo.

Una *valutazione della funzione obiettivo* corrisponde al calcolo del numero di colori utilizzati da una colorazione completa e ammissibile del grafo.

Nella relazione finale devono essere riportati:

- il numero di colori ottenuto in ciascuna run;
- l'assegnazione dei colori della run migliore;

- la verifica dell'ammissibilità della soluzione;
- 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. **Local Search (LS)**, con 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 soluzione iniziale casuale, vicinato basato sul riassegnamento del colore di un nodo e schema di raffreddamento lasciato alla scelta del candidato. Tutti i parametri devono essere dichiarati e motivati nella relazione.
4. **Algoritmo Genetico (GA)**, con codifica della soluzione come vettore di colori, operatori di selezione, crossover e mutazione lasciati alla scelta del candidato. È obbligatorio garantire l'ammissibilità di 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 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, 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 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 **18 maggio 2026**. Il candidato è invitato a comunicare la scelta dell'algoritmo da sviluppare entro le ore 16:00 del **1 maggio 2026**.