

# 1 Ricerca

## 1.1 Criteri di valutazione delle strategie

1. **Completezza**

Garanzia di trovare una soluzione, se esiste

2. **Ottimalità**

Garanzia di trovare una soluzione ottima (a costo minimo)

3. **Complessità Temporale**

Quanto tempo occorre per trovare una soluzione

4. **Complessità Spaziale**

Quanta memoria occorre per effettuare la ricerca

### 1.1.1 Valutazione della Complessità (Temporale e Spaziale)

Si usa Tempo e Spazio **parametrici**. Calcolati in termini di **numero di nodi creati o visitati**.

## 1.2 Strategie di Ricerca

Ci sono due tipologie di approcci (riguardanti la conoscenza, quanta e quale)

1. **Approcci Blind**

usano esclusivamente la struttura del problema per cercare (e trovare) una soluzione.

2. **Approcci Informati**

usano la struttura del problema + ulteriore conoscenza per guidare la ricerca.

- (a) Monoagente
- (b) Multiagente (giochi)
- (c) Problemi di Assegnamento (**CSP**)

### 1.2.1 RICERCA IN AMPIEZZA

- La ricerca espande il nodo radice, poi tutti i suoi successori, poi tutti i discendenti di secondo livello, ecc.
- Si realizza gestendo la frontiera come una coda FIFO.
- Tutti i nodi sulla frontiera e tutti i loro antenati vanno tenuti in memoria per poter ricostruire la soluzione quando si trova il nodo obiettivo (Complessità Spaziale)

### Valutazione

**Completezza** Se esiste un nodo obiettivo a una profondità **finita**  $d$  esso verrà trovato **a patto che il fattore di ramificazione** (cioè il numero di figli che un nodo può avere)  $b$  sia finito

**Ottimalità** La soluzione trovata è ottima **solo se** il costo del cammino è **una funzione monotona crescente della profondità** (es. tutte le azioni hanno lo stesso costo, quindi *il costo* dipende solo dalla profondità dell'albero)

**Complessità temporale** di tipo **esponenziale**  $O(b^{d+1})$   
 Quanto tempo occorre per trovare una soluzione col crescere dello spazio di ricerca

**Complessità spaziale** di tipo **esponenziale**  $O(b^{d+1})$   
Quanta memoria occorre per effettuare la ricerca col crescere dello spazio di ricerca

### 1.2.2 RICERCA A COSTO UNIFORME

- Quando i costi dei passi **non sono tutti identici**:
  - ogni nodo ha associato il costo del cammino con cui è stato raggiunto
  - la frontiera è mantenuta ordinata
  - a ogni iterazione **espande il nodo appartenente a un cammino di costo minimo**
- Quando i costi sono tutti **uguali** diventa la ricerca in **ampiezza**.
- **ATTENZIONE**
  - $\text{COSTO} \neq \text{NUMERO DEI PASSI}$   
il numero dei passi non conta, conta solo il costo dei cammini
  - quando trova il nodo obiettivo **NON SI FERMA SUBITO**, prima controlla se vi sono cammini aperti di costo inferiore e, nel caso, prova ad espanderli.

### 1.2.3 Ricerca in profondità (senza BackTracking)

### 1.2.4 Iterative Deepening

### 1.2.5 Ricerca Bidirezionale

## 2 Constraint Satisfaction Problem

Questi problemi sono problemi in cui gli stati assumono una **struttura** e **contengono informazioni** che vengono utilizzate nella ricerca della soluzione.

Viene anche introdotta la nozione di **vincolo** e nuovi metodi di ricerca specifici.

Un **CSP** è definito da:

- un insieme di **variabili**  $X_1, \dots, X_n$
- un insieme di **vincoli**  $C_1, \dots, C_m$
- (OPZIONALE) la richiesta di **massimizzare** una **funzione obiettivo**.

Le **variabili** prendono i valori tramite l'**assegnamento**, che è un'attribuzione di valori ad alcune variabili del nostro problema

$$\{X_{i1} = v_{i1}, X_{i2} = v_{i2}, \dots\}$$

Un **assegnamento** è detto:

- **COMPLETO** se assegna valori a tutte le variabili del CSP
- **CONSISTENTE** se non viola alcun vincolo del CSP
- **SOLUZIONE** se è completo e consistente