

Problemi di pianificazione

Dato un agente in grado di compiere delle azioni nel mondo in cui è immerso, un problema di pianificazione, riguarda la scelta di un piano di azioni, che l'agente può compiere al fine di portare il mondo dal suo stato attuale allo stato di goal.

È possibile definire un problema di pianificazione come una quintupla:

$$\langle P, A, I, G, \tilde{G} \rangle$$

dove:

- $P \neq \emptyset$ è un insieme finito di proposizioni;
- $A \neq \emptyset$ è un insieme finito di azioni, composto da una quintupla $\langle n, R, \tilde{R}, T, \tilde{T} \rangle$:
 - n è un simbolo che identifica il nome dell'azione;
 - R è un sottoinsieme di P detto preconditione asserita;
 - \tilde{R} è un sottoinsieme di P detto preconditione negata;
 - T è un sottoinsieme di P detto postcondizione asserita;
 - \tilde{T} è un sottoinsieme di P detto postcondizione negata;
- $I \neq \emptyset$ è un sottoinsieme finito di P detto stato iniziale;
- $G \neq \emptyset$ è un sottoinsieme finito di P detto goal asserito;
- $\tilde{G} \neq \emptyset$ è un sottoinsieme finito di P detto goal negato;

Un'azione $a \in A$, con $a = \langle n, R, \tilde{R}, T, \tilde{T} \rangle$, può essere compiuta dall'agente se il mondo si trova in uno stato S tale che:

$$R \subseteq S \wedge \tilde{R} \cap S = \emptyset$$

In ogni istante, il mondo è caratterizzato da uno stato.

Per risolvere un problema di pianificazione è possibile operare una **ricerca nello spazio degli stati** che parta dallo stato iniziale I e proceda all'identificazione delle soluzioni al problema, dove i nodi vengono organizzati in un **albero di ricerca**.

Fissato un percorso nell'albero di ricerca, l'insieme dei nodi costruiti e non ancora scartati viene detto **frangia**.

Se ogni nodo viene:

- Aggiunto in fondo alla frangia, allora l'algoritmo di ricerca nello spazio degli stati viene detto **in ampiezza** (BFS Breadth-First Search), nel quale in un istante dobbiamo avere in memoria tutti i figli del nodo espanso.
- Aggiunto in fondo alla frangia, allora l'algoritmo di ricerca nello spazio degli stati viene detto **in profondità** (DFS Depth-First Search), nel quale la quantità di memoria da usare è lineare rispetto alle azioni necessarie per il goal.
- Aggiunto in modo tale che gli stati più vicini agli stati di goal siano più vicini alla testa della frangia, allora l'algoritmo di ricerca nello spazio degli stati viene detto **informato** (è una via di mezzo tra BFS e DFS).

Tutti i metodi hanno complessità computazionale asintotica di caso pessimo di esponenziale di ordine $O(b^d)$, dove $b = |A|$ e d è il numero di azioni necessarie per raggiungere, nel caso pessimo uno stato di goal. Viceversa,

1. **BFS** ha complessità computazionale spaziale asintotica di caso pessimo di ordine $O(b^d)$ e garantisce che per problemi risolubili, venga ottenuta una soluzione composta dal **numero minimo di azioni**.
2. **DFS** ha complessità computazionale spaziale asintotica di caso pessimo di ordine $O(d)$ pur *non garantendo*, che per problemi risolubili, venga ottenuta una soluzione composta dal numero minimo di azioni.

Per ottenere un buon compromesso tra BFS e DFS di solito si utilizza una **ricerca ad approfondimenti successivi** ID-DFS. Nella quale l'obiettivo è quello di tenere basso l'uso di memoria e ricercare lo stato di goal ottenibile con il numero minimo di azioni come BFS, lo svantaggio è che tutte le volte viene ricostruito l'albero.

1. Viene inizializzato il **parametro di profondità massima N** al valore 1;

2. Viene applicata la *DFS* fino alla costruzione di un albero di ricerca con non più di N livelli;
3. Se viene trovata una soluzione allora viene immediatamente ritornata senza necessariamente completare l'albero di ricerca; ma
4. Se non viene trovata una soluzione allora viene incrementato N di uno e si ritorna al punto 2.

Normalmente, la ricerca informata, richiede che:

1. Il costo sia **definito positivo** (mai nullo o negativo) e che sia **additivo**.
2. Venga definita una **funzione di costo** per ogni nodo N , che quantifichi il costo minimo necessario per partire dalla radice, passare per il nodo N e raggiungere lo stato di goal.

La ricerca nello spazio degli stati può essere:

- **in avanti** (forward chaning) → parte dallo stato iniziale fino ad arrivare ad uno stato di goal.
- **all'indietro** (backward chaning) → parte dallo stato di goal fino ad arrivare allo stato iniziale, invertendo il ruolo delle postcondizioni con le precondizioni.

Algoritmo Informato A*:

Algoritmo A*

In informatica, A* è un algoritmo di ricerca su grafi che individua un percorso da un dato nodo iniziale verso un dato nodo goal. Utilizza una "stima euristica" che classifica ogni nodo attraverso una stima

☞ https://it.wikipedia.org/wiki/Algoritmo_A*

