

Introduzione all'Intelligenza Artificiale – a.a. 2017/18

Prima prova intermedia
10 Aprile 2018

PARTE I - Domande (punti 7)

La risposta corretta alla maggior parte delle domande abilita la correzione della seconda parte. Fornire risposte brevi, entro una riga di testo.

1. Nella formulazione di un problema, quali componenti definiscono lo spazio degli stati?
Le componenti del problema che definiscono lo spazio degli stati sono: lo stato iniziale, le azioni possibili e il modello di transizione (di stato).
2. Qual è il principale vantaggio della ricerca in profondità?
La visita ad albero ha costo in memoria bm , oppure m per la ricerca ricorsiva, dove b è il fattore di diramazione, m la lunghezza massima del cammino nello spazio degli stati. Quindi il vantaggio principale sta nella minore occupazione di memoria.
3. Esprimere il costo (complessità asintotica) della UC (ricerca a costo uniforme) nelle condizioni in cui sia una ricerca ottimale
Con costo arco $\epsilon > 0$, $O(b^{1+\lceil C^/\epsilon \rceil})$, dove C^* è il costo della soluzione ottima*
4. Euristiche: quale euristica sarebbe da preferire in un labirinto in cui sono ammesse mosse in orizzontale, verticale e diagonale e ogni mossa costa 1?
a. Distanza Manhattan
b. Distanza euclidea approssimata all'intero per eccesso
c. Distanza euclidea approssimata all'intero per difetto
Né la MD, né la distanza euclidea approssimata per eccesso sono euristiche ammissibili in questo contesto, quindi la risposta giusta è la c, che rappresenta una euristica ammissibile per il problema.
5. Ci sono casi in cui MIN-MAX e Alfa-Beta portano a scegliere mosse diverse?
NO, a parità di limite alla profondità di esplorazione.
6. In un problema CSP ad ogni iterazione si sceglie una variabile da assegnare: quali euristiche sono previste per questa scelta?
L'euristica MRV (Minimum Remaining Values), che sceglie la variabile con meno valori residui nel dominio, e l'euristica del grado, la variabile coinvolta in più vincoli con le altre variabili.
7. Dare la definizione di completezza di un sistema deduttivo.
 $KB \models A \text{ implica } KB \models A$ per ogni A .

PARTE II - Esercizi

Esercizio 1. Algoritmi di ricerca (punti 10)

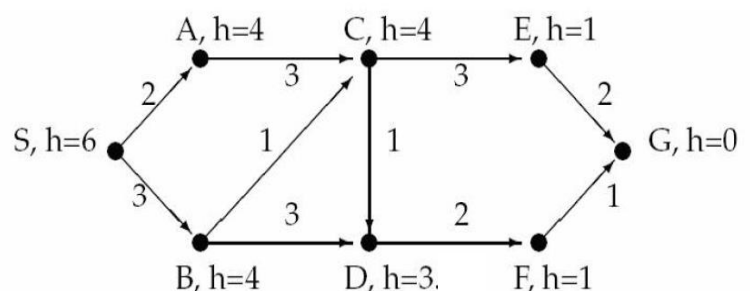
Nel grafo a fianco è rappresentato uno spazio di ricerca con S, stato iniziale e G stato goal.

Vi chiediamo di tracciare l'andamento di alcuni algoritmi mostrando ad ogni passo la coda di priorità che gestisce la frontiera.

A parità di euristica si segue l'ordine alfabetico.

- a. Greedy Best First.

Andamento dei nodi sulla frontiera, col progredire



dell'algoritmo.

<pre> graph TD S["S [6]"] --> A["A [4]"] S --> B["B [4]"] A --> C1["C [4]"] B --> C2["C [4]"] B --> D["D [3]"] D --> F["F [1]"] F --> G["G [0]"] </pre>	Step	Nodo espanso	Frontiera
	1	S	A[4], B[4]
	2	A	C[4], B[4]
	3	B	C[4], D[3]
	4	D	C[4], F[1]
	5	F	C[4], G[0]
	6	G	goal trovato

Cammino soluzione: SBDFG di costo 9. [Il cammino ottimo è SBCDFG di costo 8]

b. A* nella versione grafo (i numeri sugli archi rappresentano i costi).

<pre> graph TD S["S [0+6]"] --> A["A [2+4]"] S --> B["B [3+4]"] A --> C1["C [5+4]"] B --> C2["C [4+4]"] B --> D1["D [6+3]"] C2 --> E["E [7+1]"] C2 --> D2["D [5+3]"] E --> G1["G [9+0]"] D2 --> F["F [7+1]"] F --> G2["G [8+0]"] </pre>	Step	Nodo espanso	Frontiera
	1	S	A[6], B[7]
	2	A	C[9], B[7]
	3	B	C[8], D[9]
	4	C	E[8], D[8]
	5	D	F[8], E[8]
	6	E	G[9], F[8]
	7	F	G[8]
	8	G	goal trovato

Cammino soluzione: SBCDFG di costo 8. Ottimo.

c. A* nella versione grafo ponendo $h(E)$ (la h sul nodo E) = 3

	Step	Nodo espanso	Frontiera
	1	S	A[6], B[7]
	2	A	C[9], B[7]
	3	B	C[8], D[9]
	4	C	E[10], D[8]
	5	D	F[8], E[10]
	6	F	G[8]
	7	G	goal trovato

Cammino soluzione: SBCDFG di costo 8

Nonostante l'euristica non sia monotona in questo caso si trova comunque la soluzione ottima (anche con un passo di espansione in meno).

d. Si trova una soluzione ottima in entrambi i casi? Se no, perché?

Si trova in entrambi i casi una soluzione ottima.

Esercizio 2. UC e A* (punti 5)

Dato il grafo in figura (con indicati i costi sugli archi):

a. Determinare il cammino soluzione (da a a g) trovato con UC nella versione ricerca-grafo e il suo costo.

Il cammino trovato è A-C-G con costo 20.

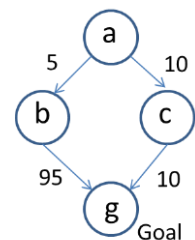
b. Quale parte dell'algoritmo ha permesso di trovare una soluzione ottima o cosa lo ha impedito?

Si è ottimale. Perché viene operato uno scambio sulla frontiera. Dall'algoritmo: **if figlio.Stato è in frontiera con $g(n)$ più alto then sostituisci quel nodo frontiera con figlio.**

c. Se usiamo A* su grafo (invece di UC) con euristica monotona (h consistente) cambiano (e come) le risposte alle domande dei punti a e b? Perché?

No, non cambiano, perché le proprietà della f monotona riguarda gli esplorati, quelli selezionati per l'espansione e non la frontiera (si veda anche la dimostrazione, cammino ottimale selezionando per l'espansione, non quando si genera e si inserisce in frontiera).

Risposta alternativa: Si può notare che anche h di UC è monotona, se bastasse la consistenza di h anche UC non utilizzerebbe l'if finale.



Esercizio 3. DPLL (punti 10)

Data la seguente KB:

- Si trasformi la KB in forma a clausole
- Si mostri con DPLL se la proposizione T è o meno conseguenza logica della KB.
- Vero o falso che DPLL è un algoritmo completo per la soddisfacibilità?

$V \vee T$
 $P \wedge \neg U$
 $R \vee \neg Q$ **KB**
 $V \Rightarrow W$
 $P \Rightarrow Q$
 $S \Rightarrow (U \vee T)$
 $(P \wedge R) \Rightarrow S$

d. L'utilizzo di WalkSAT in questo caso (punto b) sarebbe consigliato/sconsigliato e perché?

a. Forma a clausole di KB.

1. {V, T}
2. {P}
3. {¬U}
4. {R, ¬Q}
5. {¬V, W}
6. {¬P, Q}
7. {¬S, U, T}
8. {¬P, ¬R, S}

b. $KB \models T$?

Provo a dimostrare che $KB \cup \{\neg T\}$ è insoddisfacibile. Questa è una possibile traccia di DPLL.

<ol style="list-style-type: none"> 1. {V, T} 2. {P} 3. {¬U} 4. {R, ¬Q} 5. {¬V, W} 6. {¬P, Q} 7. {¬S, U, T} 8. {¬P, ¬R, S} 9. {¬T} goal negato <p>Clausola con letterale puro: W= True. La 5 è soddisfatta.</p>	<ol style="list-style-type: none"> {V, T} {P} {¬U} {R, ¬Q} {¬P, Q} {¬S, U, T} {¬P, ¬R, S} {¬T} <p>Clausole unitarie: P=True. Le clausole che contengono P sono soddisfatte. Le clausole che contengono ¬P si semplificano.</p>	<ol style="list-style-type: none"> {V, T} {¬U} {R, ¬Q} {Q} {¬S, U, T} {S} {¬R, S} {¬T} <p>Clausola unitaria. U=False</p>
<ol style="list-style-type: none"> {V, T} {R, ¬Q} {Q} {¬S, T} {S} {¬R, S} {¬T} <p>Clausola unitaria: Q = True</p>	<ol style="list-style-type: none"> {V, T} {R} {¬S, T} {S} {¬R, S} {¬T} <p>Clausola unitaria: R=True</p>	<ol style="list-style-type: none"> {V, T} {¬S, T} {S} {¬T} <p>Letterale puro: V =True</p>
<ol style="list-style-type: none"> {¬S, T} {S} {¬T} <p>Clausola unitaria: S=True</p>	<ol style="list-style-type: none"> {T} {¬T} <p>Impossibile trovare un modello.</p>	

L'insieme è insoddisfacibile quindi $KB \not\models T$

c. Vero o falso che DPLL è un algoritmo completo per la soddisfacibilità?
VERO.

d. L'utilizzo di WalkSAT in questo caso (punto b) sarebbe consigliato/sconsigliato e perché?

Sconsigliatissimo. Potrebbe non terminare nel tentativo di trovare un modello o terminare per avere raggiunto il numero massimo di passi, senza arrivare a una conclusione.