Euristiche ammissibili

Una funzione euristica h è detta ammissibile quando ∀ n, $h(n) \leq h^*(n)$ dove $h^*(n)$ è il costo minimo reale per raggiungere il nodo goal a partire dal nodo n

- Intuitivamente un'euristica è ammissibile quando non fa mai stime per eccesso, è ottimistica
- **Esempio**: la distanza in linea d'aria è un'euristica ammissibile rispetto alla distanza su strada

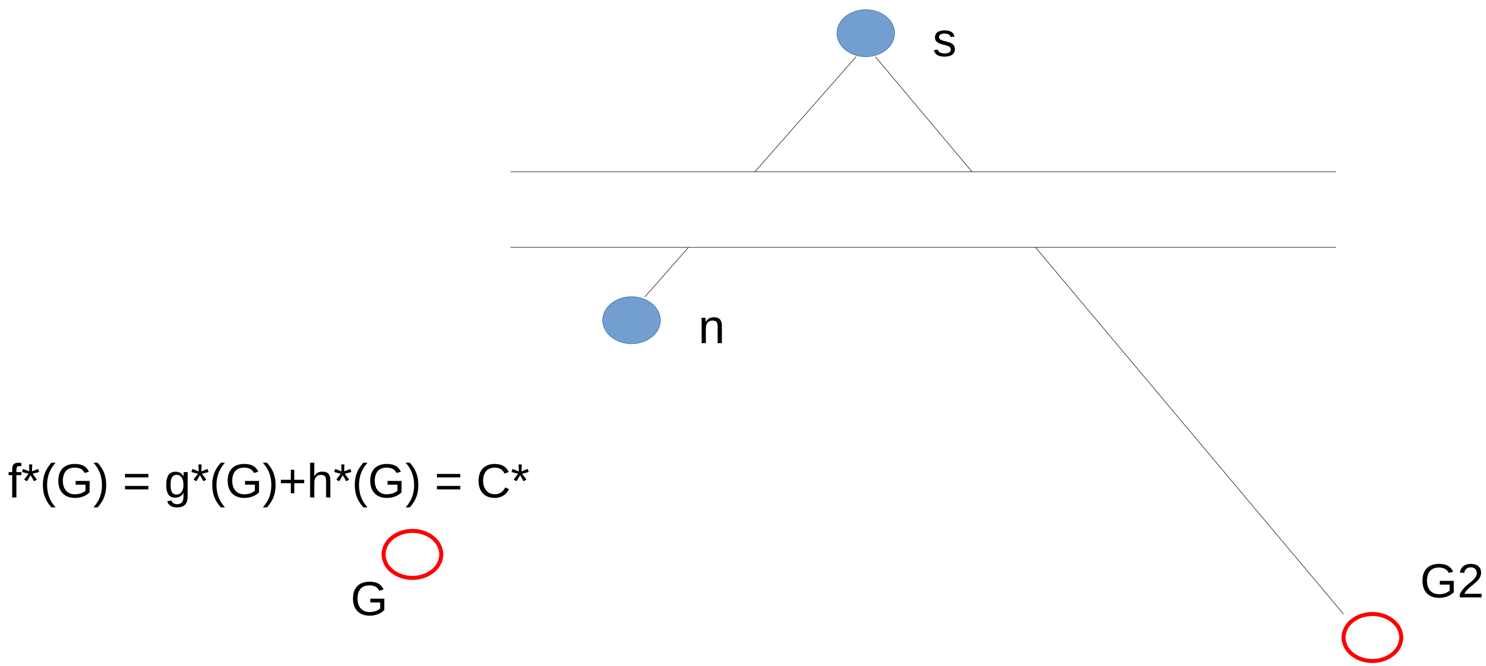
Ottimalità dell'algoritmo A*

- Si dimostra che quando:
  1) l'euristica h è ammissibile
  2) e tutti i passi hanno un costo maggiore di una costante positiva piccola a piacere
- Allora:
  - **A* termina e trova una soluzione ottima** (di costo minimo)
  - In altri termini in questo caso A* è completo e ottimale

A* ottimale per alberi di ricerca

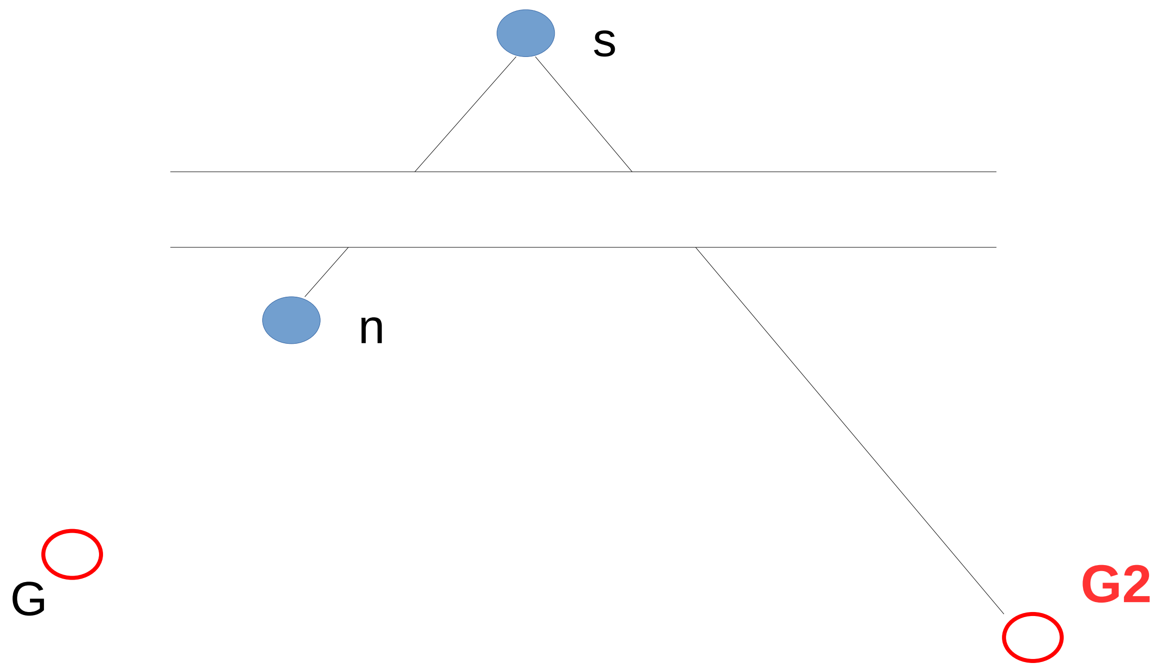
- **Nota: in un albero di ricerca ogni nodo ha un solo cammino assoluto**
- Perché manchi l'ottimalità deve accadere che durante la ricerca l'algoritmo:
  - scelga un nodo obiettivo sub-ottimo (G2)
  - al posto di un nodo (n)
  - che si trova su un cammino ottimo (che porta al goal G)
- Dimostriamo che in un albero di ricerca ciò non può capitare laddove l'euristica è ammissibile e i passi hanno costo non nullo

A* ottimale per alberi di ricerca



- Chiamiamo:
  - **G2 = obiettivo subottimo**
  - **C* il costo reale della soluzione ottima**

A* ottimale per alberi di ricerca



f(G2) = g(G2)+~~h(G2)~~ > C*

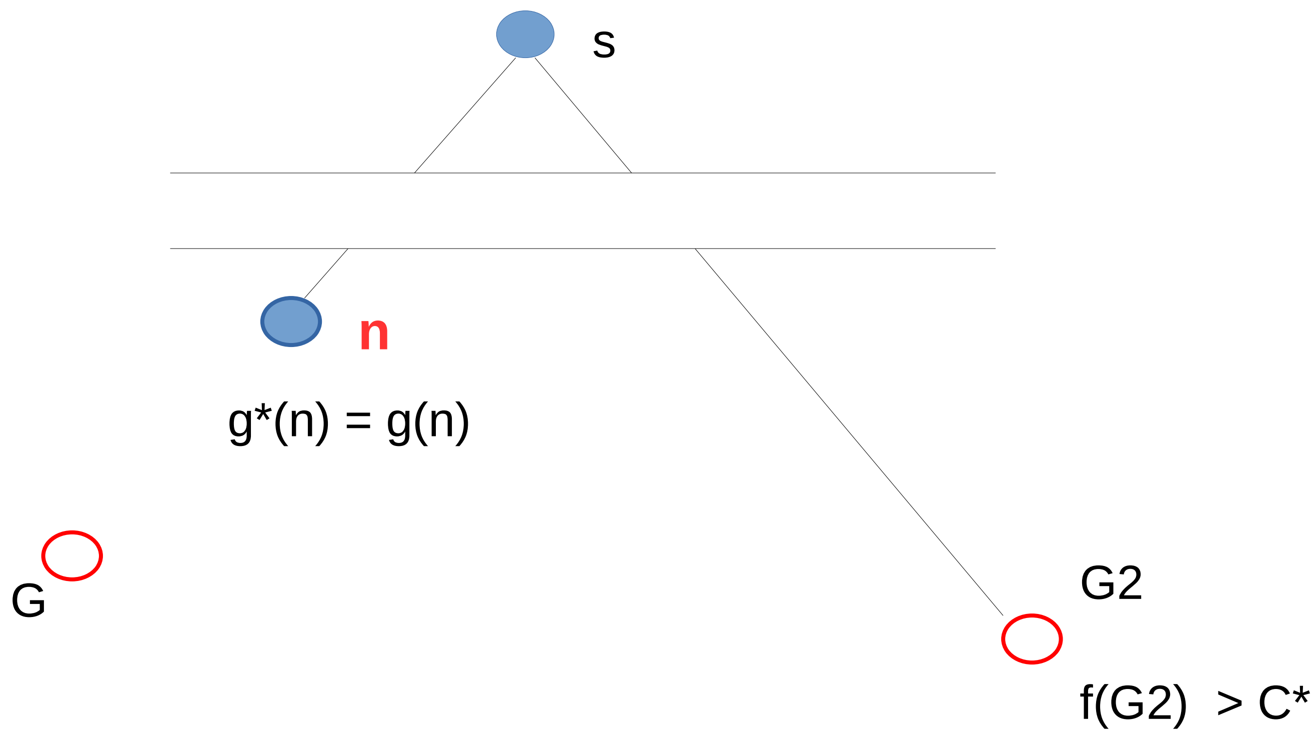
Consideriamo G2:

1) **h(G2) = 0** perché G2 è un nodo obiettivo

2) **f(G2) =** g(G2) + h(G2) = **g(G2) + 0 = g(G2)**

3) Poiché per assunto G2 è <u>subottimo</u>, si ha che:
f(G2) = g(G2) > C*

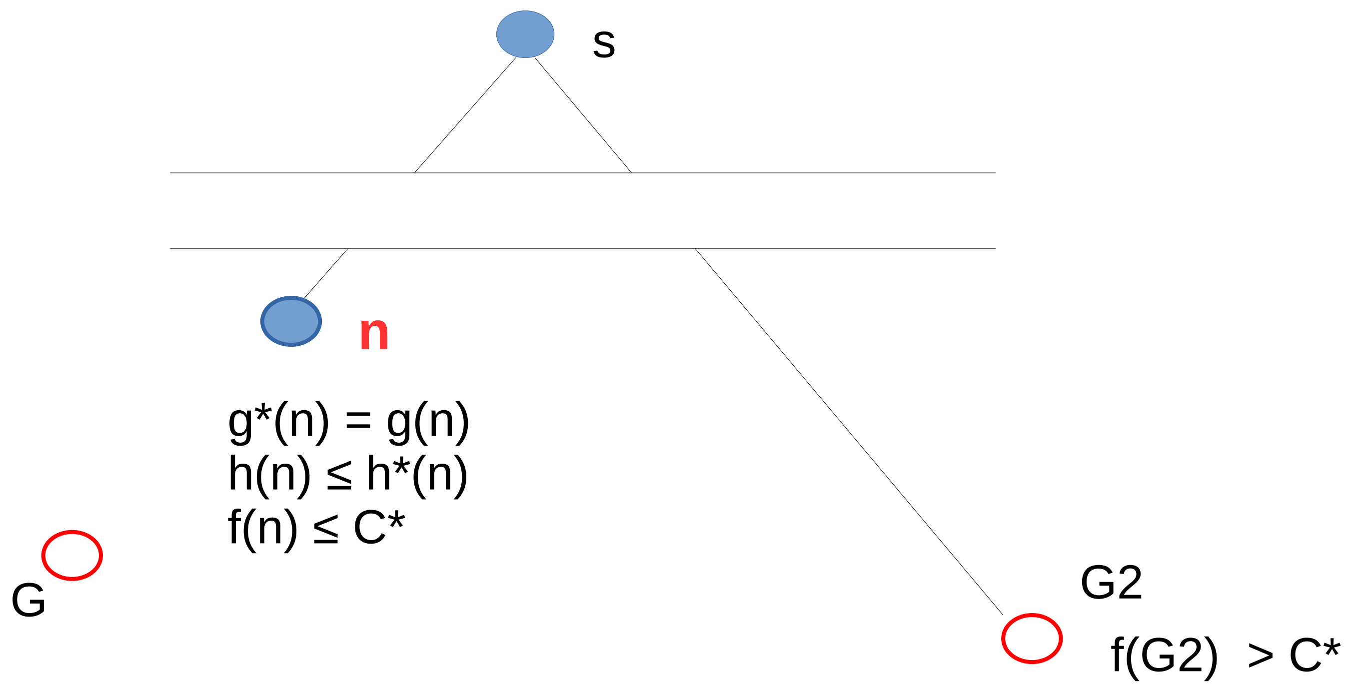
A* ottimale per alberi di ricerca



Sia **n** un **generico nodo** appartenente a un **cammino ottimo:**

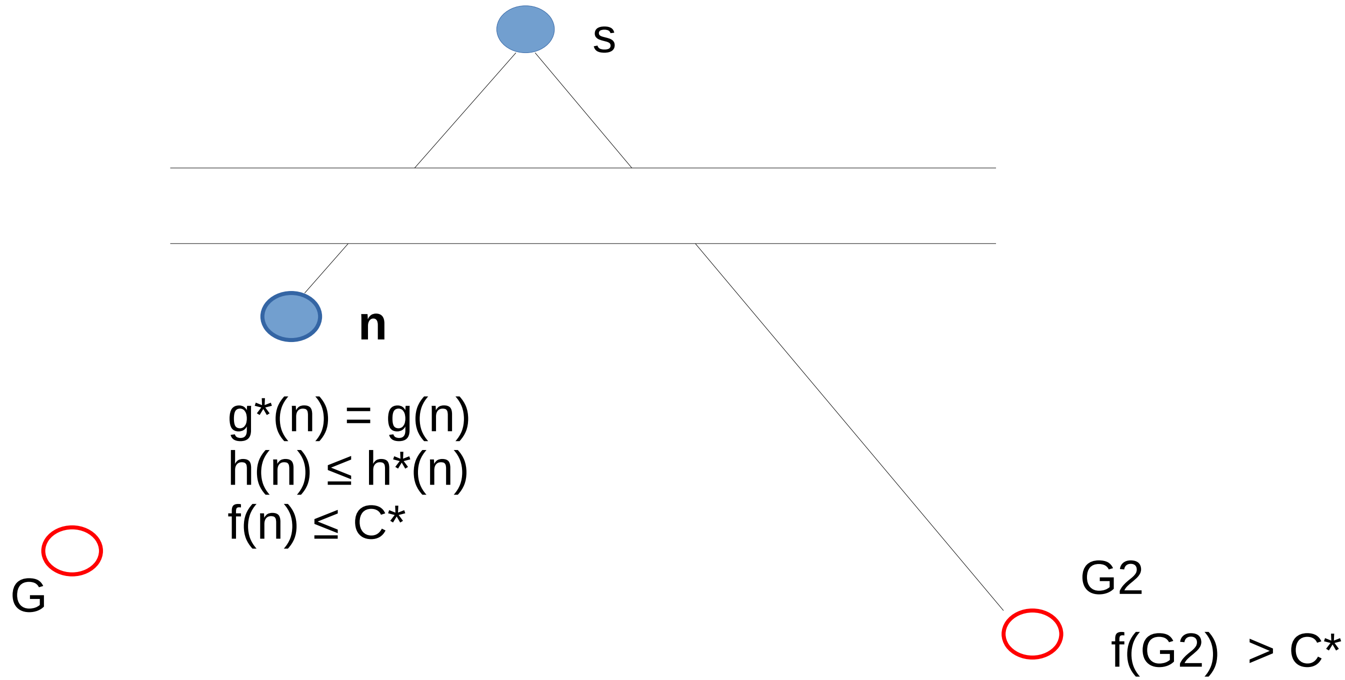
- **g*(n) = g(n)** perché lavoriamo su di un albero e ogni nodo è raggiungibile dalla radice lungo un solo percorso

A* ottimale per alberi di ricerca



- **h** è **ammissibile** per ipotesi, quindi **h(n) ≤ h*(n)**
- Quindi: **f(n) =** g(n) + h(n) **≤** g*(n) + h*(n) = **C***
- Quindi **f(n)** ≤ C* **< f(G2)**

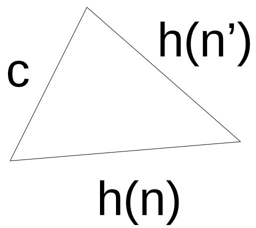
A* ottimale per alberi di ricerca



- Quindi:
  1)f(n) < f(G2)
  2)A* sceglie il nodo aperto con f(.) minima,
- Di conseguenza fra n e G2 verrà scelto n (q.e.d.)

A* ottimale per grafi di ricerca

- **Nei grafi vi è molteplicità di cammino**:
il primo cammino trovato durante la ricerca che porta a un certo stato non è necessariamente quello ottimo (nota: nell'esempio della Romania A* non si ferma quando incontra Bucarest la prima volta)
- **Questa proprietà invalida la precedente dimostrazione!**
- La dimostrazione generale è piuttosto complessa, occorre aggiungere l'ipotesi che **h** sia **monotòna** (o consistente), cioè che dati un qualsiasi nodo n e un qualsiasi suo successore n' prodotto eseguendo l'azione a in n vale che **h(n) ≤ c(n, a, n') + h(n')**
- Tale disuguaglianza è una <u>disuguaglianza triangolare</u>



A* ottimale per grafi di ricerca

- Si dimostra che quando l'**euristica è monotona**, per la disuguaglianza triangolare i **costi f(n)** lungo un cammino **sono non decrescenti**
- **A* espande i nodi in ordine <u>non decrescente</u> di f:**
  - se un nodo CHIUSO viene incontrato più volte lungo un percorso, i nuovi valori di f saranno superiori a quello calcolato la prima volta
    (cfr. Arad nell'esempio della Romania)
  - Di conseguenza sono i primi incontri con i nodi obiettivo che permetteno di individuare una soluzione ottima
- Su questo assunto si dimostra l'ottimalità

A* ottimale per grafi di ricerca

- Dimostriamo che se l'euristica è monotona, i costi di f(n) lungo un cammino sono <u>non decrescenti</u>:

1) Sia **n'** un nodo successore di **n**, per definizione:
   g(n') = g(n) + c(n, a, n')

2) Sempre per definizione:
   f(n') = **g(n')** + h(n')

3) E sostituendo g(n') :
   f(n') = **g(n) + c(n,a,n')** + h(n')

A* ottimale per grafi di ricerca

4) Applichiamo ora la disuguaglianza triangolare:

   f(n') = g(n) + **c(n, a, n') + h(n') ≥** g(n) + **h(n)**

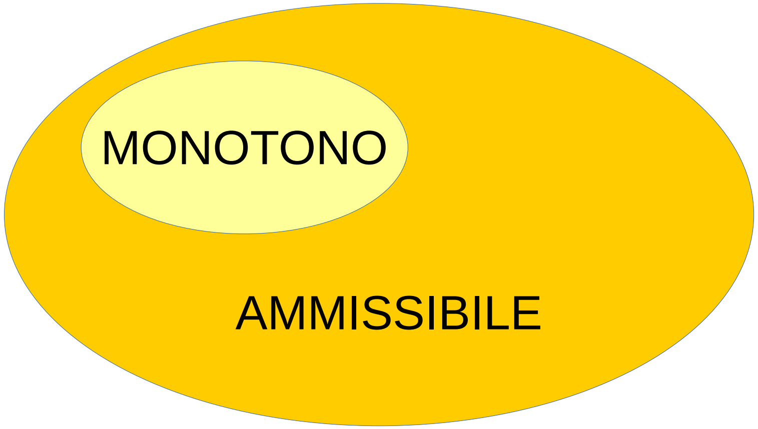
5) Sappiamo però che, per definizione: **g(n) + h(n) = f(n)**

6) Di conseguenza: **f(n') ≥ f(n)** (q.e.d.)

Euristiche e impatto sulla ricerca

Euristiche monotone e ammissibili

- La monotonicità è una proprietà più stringente dell'ammissibilità
  - Si dimostra che un'euristica monotona è anche ammissibile
  - Spesso ma non sempre le euristiche ammissibili sono anche monotone



Esempio di euristica monotona

- La **distanza in linea d'aria** è un'euristica:

  - **ammissibile** e **monotona** per il problema della Romania, infatti data una città (genericamente indicata da luogo) e un suo possibile successore (indicato da luogo1) avremo sempre che:

    h(luogo) < c(luogo, vai, luogo1) + h(luogo1)

- Cioè la distanza in linea d'aria da luogo a Bucharest è minore della distanza via terra fra luogo e il confinante luogo1 più la distanza in linea d'aria da luogo1 a Bucharest

Ammissibile non vuol dire informativo !!

- **h(n) = 0** è un'euristica sempre ammissibile ma non è informativa della desiderabilità degli stati
  - Permette di valutare solo il costo del percorso fatto per raggiungere un nodo
  - La ricerca diventa cieca e richiede l'espansione di un maggiore numero di nodi rispetto a usare un'euristica ammissibile e informativa
- In particolare se abbiamo inoltre che tutte le operazioni che permettono di passare da un nodo a un successore hanno **costo uniforme** pari a **1, A* diventa una ricerca in ampiezza**

Valutazione

- **A* è ottimamente efficiente per qualsiasi euristica**: non esiste alcun altro algoritmo ottimo che garantisca di espandere meno nodi di quelli espansi da A*
- Purtroppo il numero di nodi espansi aumenta esponenzialmente con la profondità della soluzione ottima
- A* mantiene in memoria tutti i nodi generati (è una ricerca in ampiezza)