

Strategie di ricerca informate

In cui si superano le inefficienze delle strategie di ricerca blind utilizzando opportune funzioni di valutazione della bontà (o utilità) degli stati

Domanda

Le strategie di ricerca blind non sono efficienti. E' possibile rendere la ricerca più efficiente utilizzando della conoscenza sul problema? La conoscenza permette di focalizzare la ricerca verso le direzioni più promettenti?

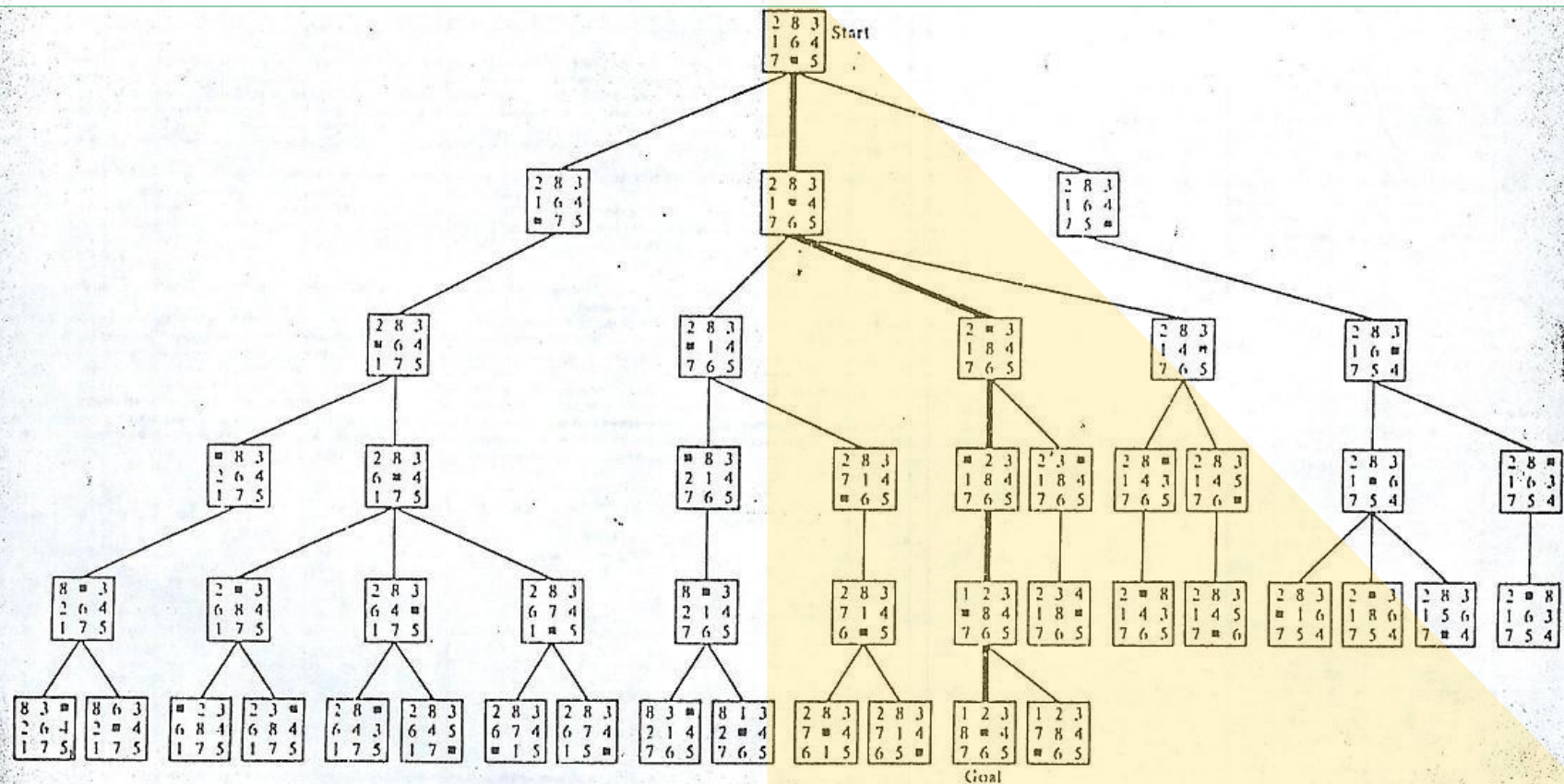


Fig. 1.4 A search tree for the 8-puzzle.

Domanda

E' possibile utilizzare conoscenza sugli stati per focalizzare la ricerca dell'obiettivo?

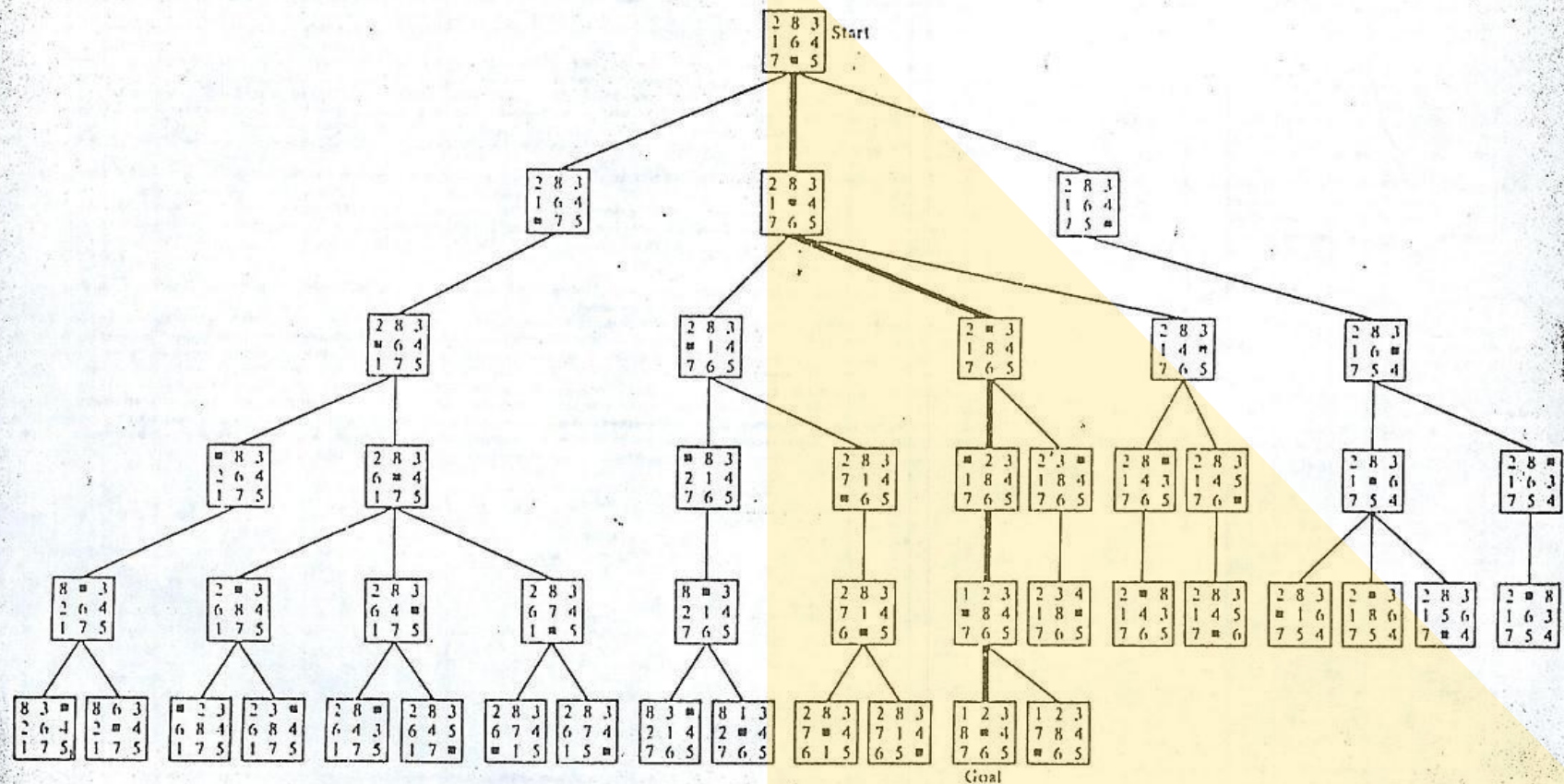


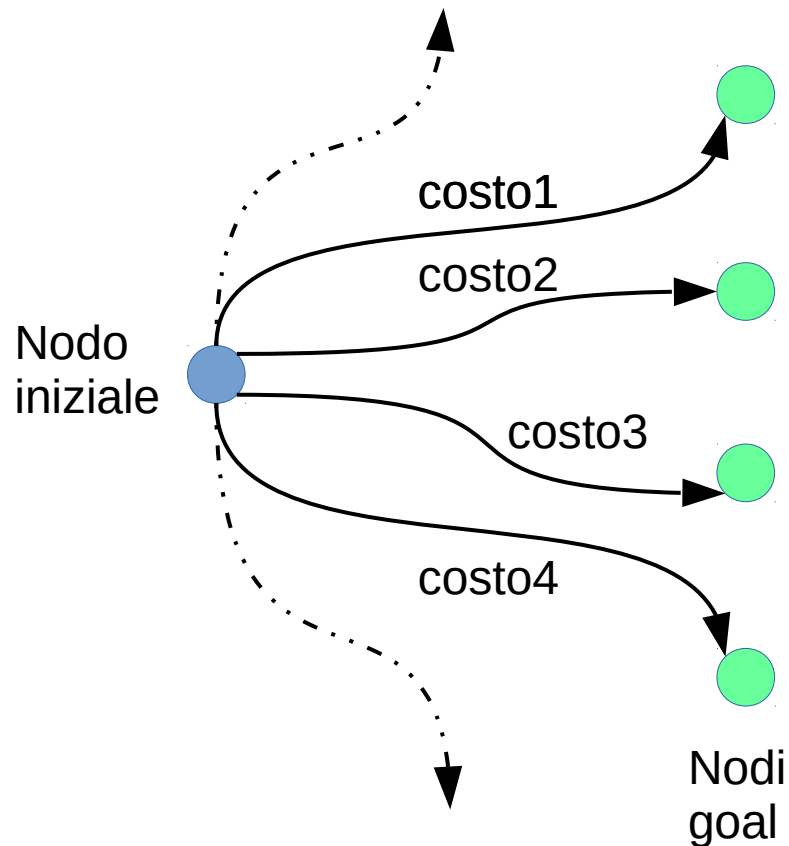
Fig. 1.4 A search tree for the 8-puzzle.

Esempio: numero di tessere al loro posto

Il contesto: grafi di ricerca

- Un grafo di ricerca è una coppia costituita da due insiemi, un insieme di nodi e un insieme di archi: $G = (\{n_i\}, \{e_{ij}\})$
- n_i : nodo
- e_{ij} : arco da n_i a n_j , la sua esistenza non implica che esista anche e_{ji}
- Ogni arco e_{ij} ha un costo c_{ij}
- Grafo dato implicitamente tramite un operatore successore gamma Γ , che restituisce i nodi successore di quello dato con i costi dei relativi archi: $\Gamma(n_i) = \{(n_i, c_{i1}), \dots, (n_i, c_{ik})\}$
- G_s indica il grafo sviluppato applicando ricorsivamente Γ a partire dal nodo iniziale s
- Cammino (n_1, n_2, \dots, n_k) insieme ordinato di nodi in cui n_{i+1} è un successore di n_i
- Costo di un cammino: somma dei costi degli archi che lo compongono

Il problema



Si vuole determinare una soluzione che permetta di raggiungere, da un nodo iniziale dato, uno fra un insieme di nodi obiettivo

I passi, e quindi i percorsi, hanno un costo

Si vuole determinare un percorso di costo minimo in modo efficiente, cioè evitando per quanto possibile di espandere nodi che non portano a questo risultato

Se esiste almeno un cammino che collega il nodo iniziale a un nodo goal allora l'algoritmo deve restituire una soluzione

Per **soluzione** si intende il percorso che collega i due nodi

Frontiera: insieme dei nodi generati ma non ancora espansi

Funzioni di valutazione e funzioni euristiche

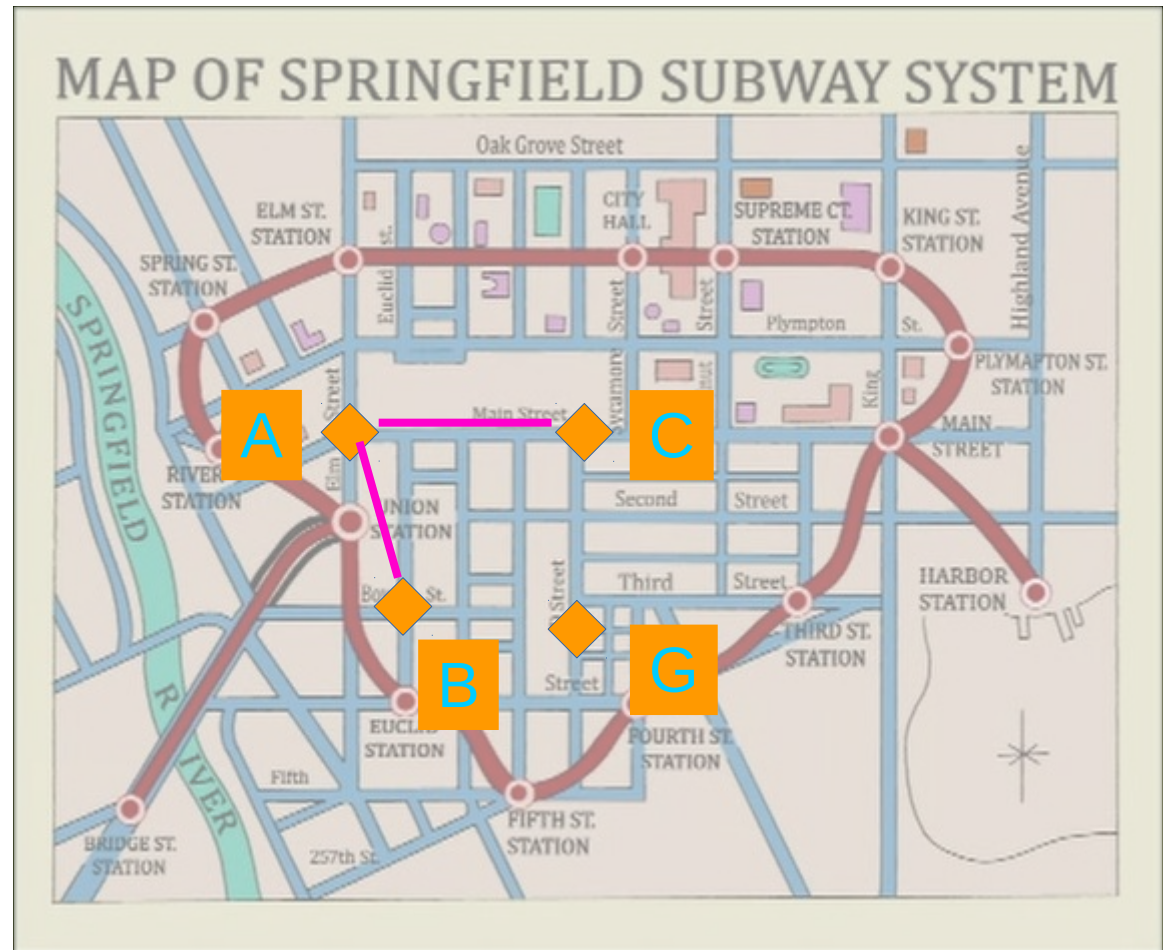
- Una strategia di ricerca informata ordina la frontiera sulla base di una funzione di valutazione $f(n)$ applicata ai nodi:
 - A seconda di come è definito $f(n)$ si ottengono strategie differenti
 - Spesso $f(n)$ comprende una componente $h(n)$ che restituisce una *stima del minimo fra i costi dei percorsi che congiungono lo stato corrispondente al nodo n a uno stato goal*
 - $h(n)$ è detta euristica
- La strategia generale è detta best-first search in quanto verrà espanso per primo lo stato “più promettente”
- Si tratta di una famiglia di strategie che comprende tutte quelle che studieremo (greedy, A^* , RBFS)

1. Ricerca greedy (avida)

- $f(n) = h(n)$
- Viene scelto il nodo stimato più vicino a quello obiettivo
 - Esempio: se gli stati sono località delle quali si hanno le coordinate, è possibile calcolare la distanza in linea d'aria tra due stati e usarla come stima della vicinanza (su strada) fra due località
- Problemi?

1. Ricerca greedy (avida)

- **Problemi?**
- In linea d'aria:
 - $(B, G) = 5$
 - $(C, G) = 7$
 - quindi $(B, G) < (C, G)$
- ma seguendo le strade:
 - $(B, G) = 10$
 - $(C, G) = 7$
 - Quindi $(B, G) > (C, G)$
- Inoltre: Rischio di loop per vicoli ciechi
- Difetti della ricerca in profondità



1. Ricerca greedy esempio



1. Ricerca greedy ancora peggio

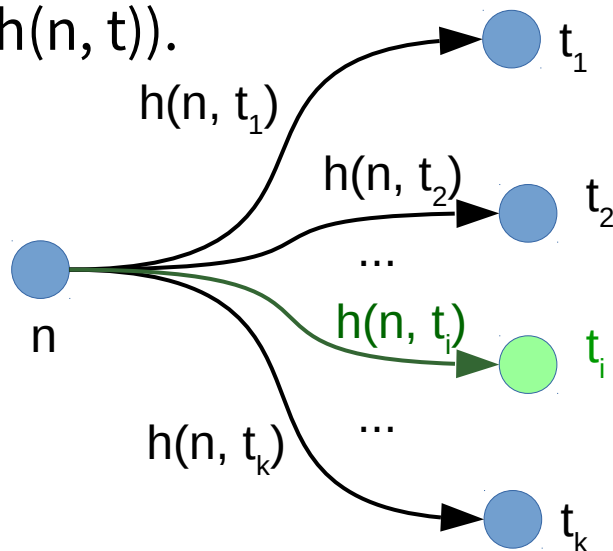
Balme – Noasca:
linea d'aria: 18Km
Su strada: 91Km



È un vicolo cieco, costringe a tornare indietro lungo il percorso di ricerca

A* e obiettivi preferiti

- A* considera grafi, generati a partire da un singolo nodo iniziale, in cui un sottoinsieme **T** di nodi è costituito da **nodi obiettivo** (T sta per target).
- Dato un qualsiasi nodo n del grafo, un obiettivo t è detto *preferito per n* se e solo se il costo del cammino ottimo $h(n, t) \leq$ costo di qualsiasi altro cammino da n verso qualsiasi altro nodo di T .
- Si indica con $h(n)$ il costo di questo cammino ottimo quindi $h(n) = \min_{t \in T} (h(n, t))$.



- Tutti i nodi $t_j \in T$
- $\forall t_j \neq t_i \ h(n, t_j) \geq h(n, t_i)$
- $h(n, t_i)$ è indicato con $h(n)$

A* e obiettivi preferiti

- A* considera grafi, generati a partire da un singolo nodo iniziale, in cui un sottoinsieme **T** di nodi è costituito da **nodi obiettivo** (T sta per target).
- Dato un qualsiasi nodo n del grafo, un obiettivo t è detto *preferito per n* se e solo se il costo del cammino ottimo $h(n, t) \leq$ costo di qualsiasi altro cammino da n a qualsiasi altro nodo di T .
- Si indica con $h(n)$ il costo di questo cammino ottimo quindi $h(n) = \min_t (h(n, t))$.
- In altri termini un nodo obiettivo è **preferito per n quando è uno dei nodi obiettivo più economici da raggiungere da n** .
- Prima vediamo lo pseudocodice di A* poi vediamo com'è fatta la funzione di valutazione

A* pseudocodice

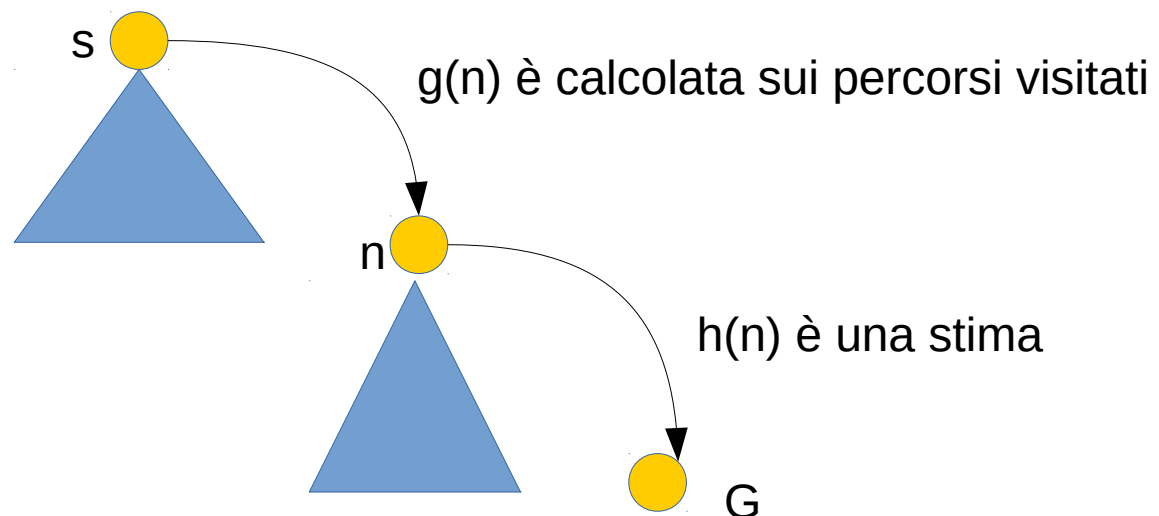
- Sia s il nodo iniziale
- Sia T l'insieme dei nodi obiettivo (target)
- segna s come **APERTO** e calcola $f(s)$;
- seleziona il nodo **APERTO** n avente valutazione minima (i pari merito dovrebbero essere risolti a vantaggio degli eventuali $t \in T$);
- se n appartiene a T , marca n **CHIUSO** e termina;
- altrimenti marca n **CHIUSO** e applica l'operatore successore Γ a n ; poi calcola il valore di f per tutti i successori n' e:
 - marca come **APERTI** quei successori n' che non risultano già **CHIUSI**;
 - rimarca come **APERTI** quei successori n' che erano **CHIUSI** ma per cui è stato calcolato un valore f più basso di quello calcolato in precedenza (cioè sono stati raggiunti tramite un percorso migliore)

Discussione su A^*

- L' algoritmo del lucido precedente è la traduzione italiana fedele dell' originale proposto da Hart, Nilsson e Raphael
- Tuttavia, qualcuno ha osservato in classe, manca un passo: se n' è un nodo APERTO che viene reincontrato e il valore $f(n')$ calcolato al nuovo giro è inferiore a quello calcolato in precedenza, bisognerebbe aggiornare $f(n')$ ed io concordo con questa osservazione
- I casi da trattare per ciascun successori n' sono:
 - 1) n' non era ancora stato incontrato: diventa APERTO ed ha assegnato il valore $f(n')$ calcolato
 - 2) n' era già stato incontrato ed è APERTO: se il nuovo valore $f(n')$ è minore di quello calcolato in precedenza tale valore viene aggiornato, altrimenti nulla cambia
 - n' era già stato incontrato ed è CHIUSO: se il nuovo valore $f(n')$ è minore di quello calcolato in precedenza tale valore viene aggiornato e il nodo torna APERTO, altrimenti nulla cambia

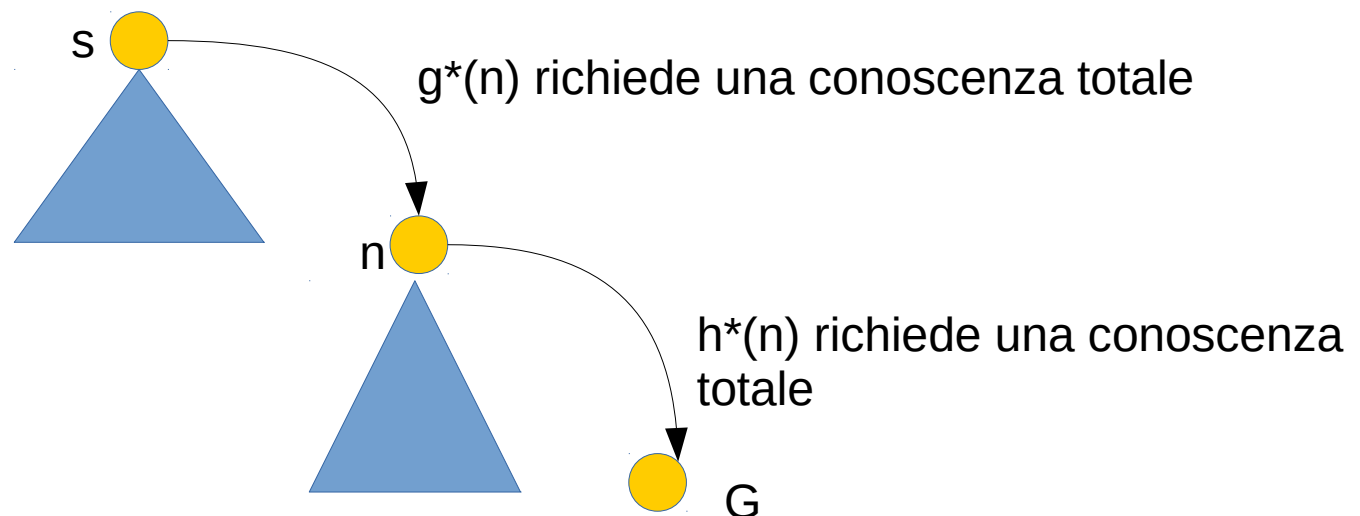
2. Ricerca A* (a star)

- $f(n) = g(n) + h(n)$
 - $g(n)$ = costo minimo di tutti i percorsi, visti fino ad ora, che consentono di raggiungere il nodo n a partire dallo stato iniziale s
 - $h(n)$ = stima del costo minimo del proseguimento di percorso che consente di raggiungere un goal preferito di n
 - $f(n)$ = stima del costo minimo per raggiungere un goal preferito di n partendo da s



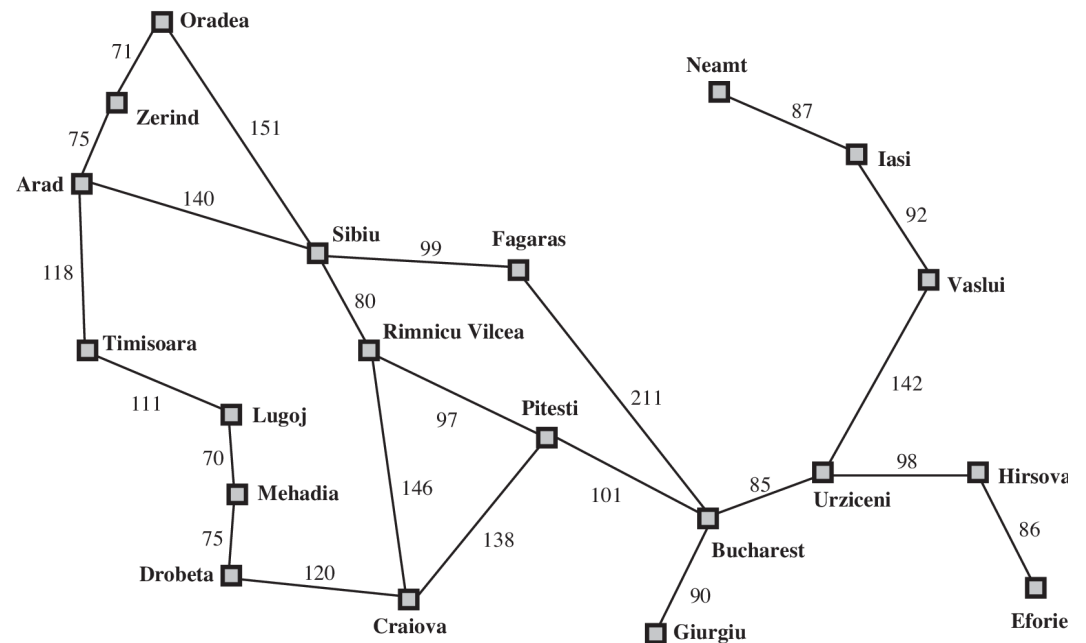
f come stima di f*

- $f^*(n) = g^*(n) + h^*(n)$
 - costo vero del cammino risolutore ottimo (nel caso esista)
 - $g^*(n)$ = costo minimo per raggiungere il nodo n a partire dallo stato iniziale s (calcolato considerando tutti i cammini possibili)
 - $h^*(n)$ = costo minimo reale del proseguimento di percorso che consente di raggiungere un goal preferito a partire dal nodo n
 - $f^*(n)$ = costo minimo per raggiungere un goal preferito di n da s



A*: esempio della Romania

- Si consideri la seguente mappa semplificata della Romania e si supponga di dover andare da Arad a Bucharest
- La mappa contiene le distanze fra le varie città collegate direttamente da strade (esempio: distanza fra Sibiu e Fagaras è 99Km)
- È possibile spostarsi da una città a una vicina e il costo è dato dai Km percorsi



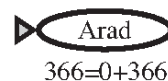
Esempio della Romania: euristica

- Sia l' euristica la distanza in linea d' aria fra ciascuna città e la destinazione Bucharest (misure riportate in tabella)
- Si noti che la distanza in linea d' aria è una stima ottimistica

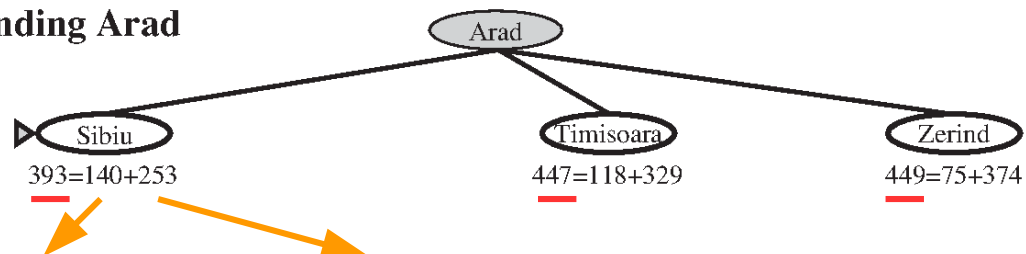
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

Uso di A* 1/3

(a) The initial state



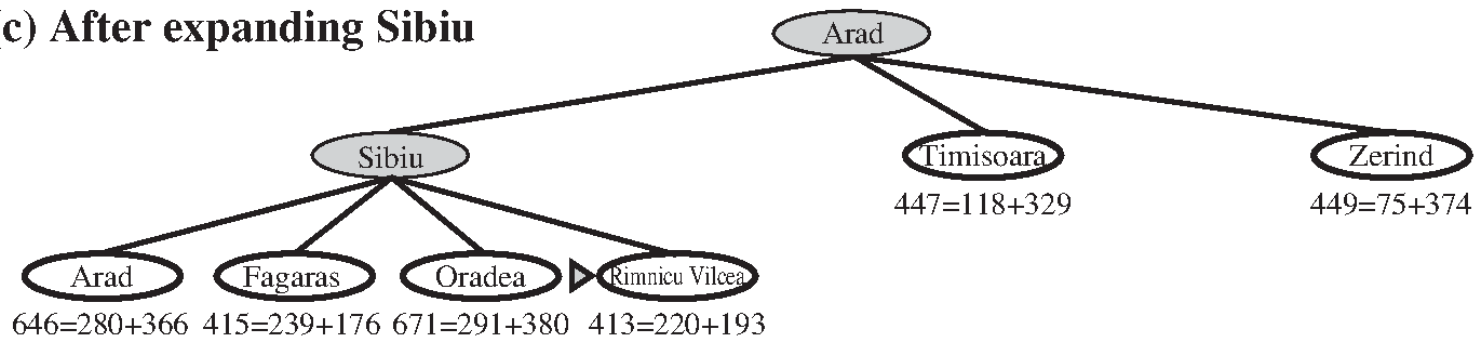
(b) After expanding Arad



Distanza su strada fra Arad e Sibiu

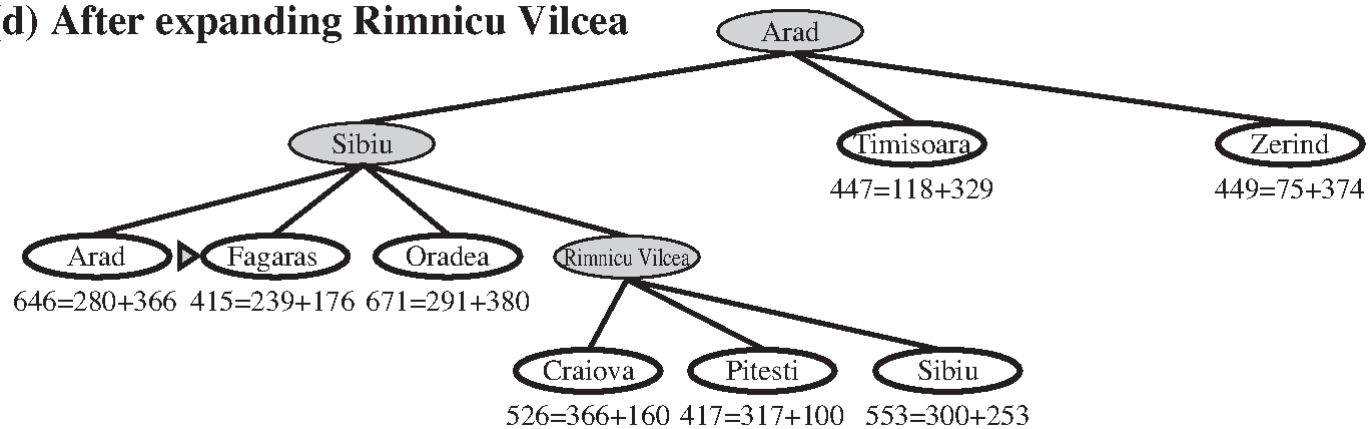
Distanza linea d'aria fra Sibiu e Bucharest

(c) After expanding Sibiu

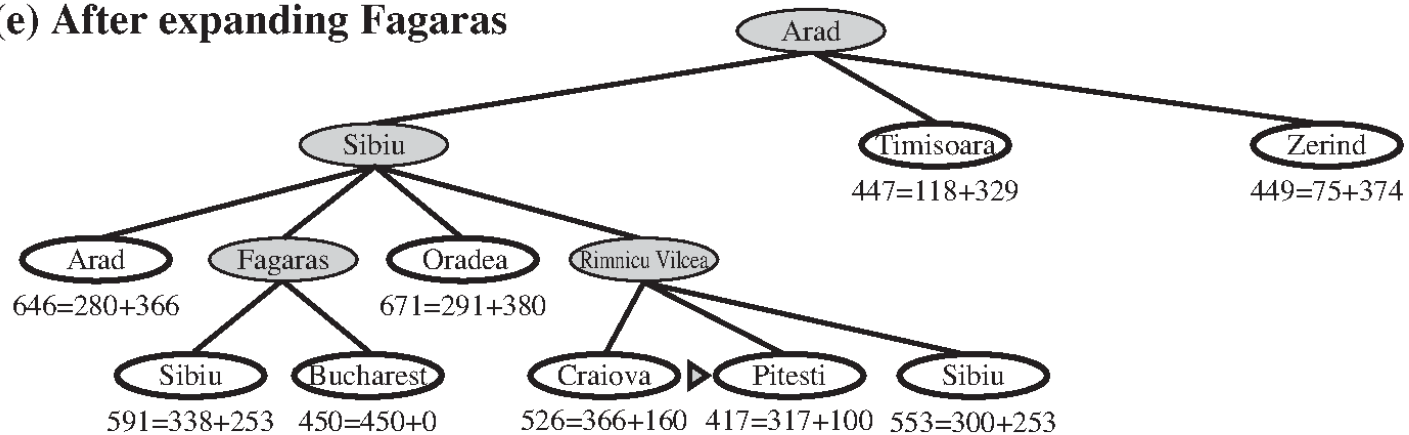


Uso di A* 2/3

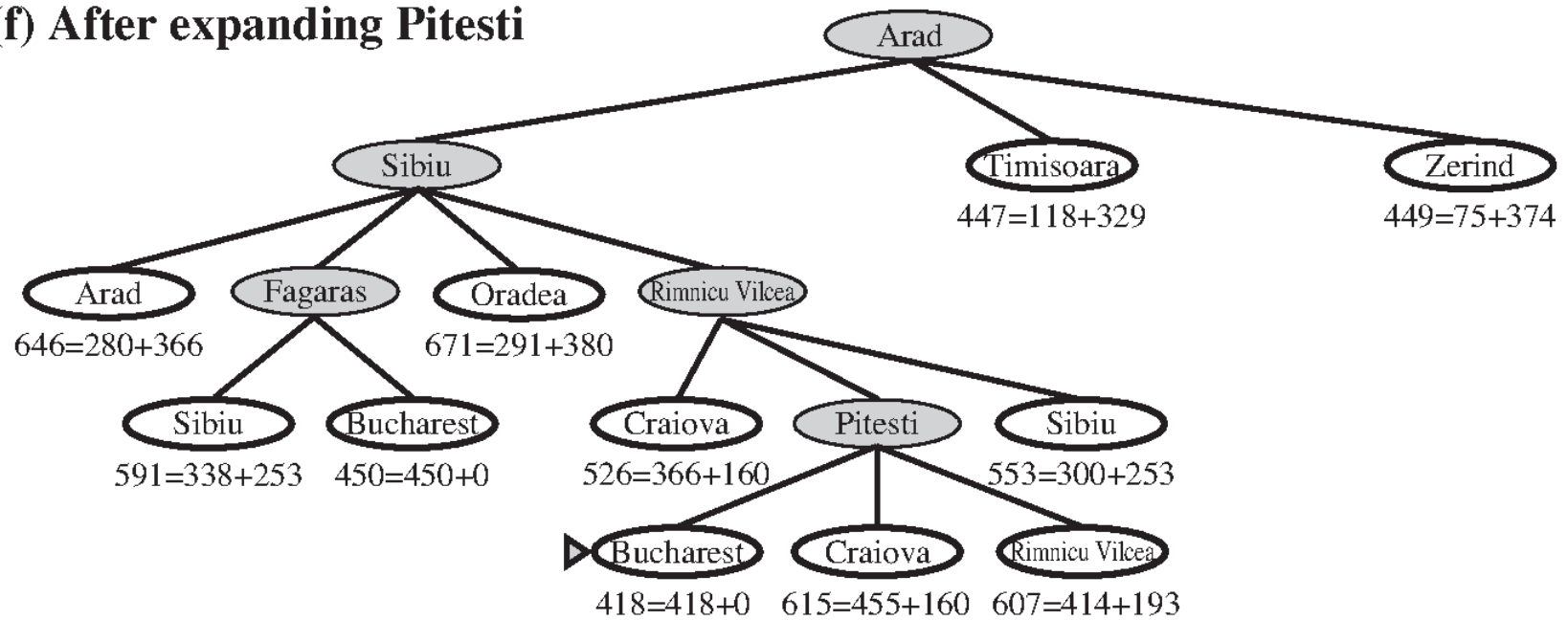
(d) After expanding Rimnicu Vilcea



(e) After expanding Fagaras



(f) After expanding Pitesti



Euristiche ammissibili

Una funzione euristica h è detta **ammissibile** quando $\forall n$, $h(n) \leq h^*(n)$ 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