Algoritmi *online*

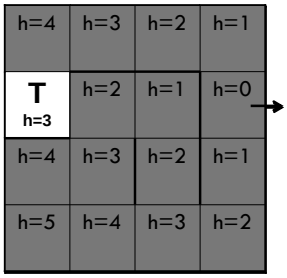
Maria Simi, a.a. 2005/06

Problemi di esplorazione

- Gli agenti per il problem-solving assumono:
  - ambienti deterministici e osservabili
  - il piano generato può essere eseguito senza problemi
- Che cosa succede se rilasciamo queste assunzioni?
  - Solo lo stato corrente è osservabile, le azioni sono stocastiche
  - Gli stati futuri e le azione che saranno possibili non sono conosciute a priori

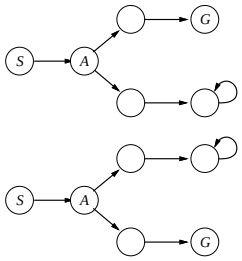
Esempio: Teseo con mappa e senza

- Con mappa
  - applicabili tutti gli algoritmi di pianificazione visti
- Senza mappa
  - l'agente non può pianificare può solo esplorare nel modo più razionale possibile
  - Ricerca online



Assunzioni

- Ambienti esplorabili in maniera sicura
  - Non esistono azioni irreversibili

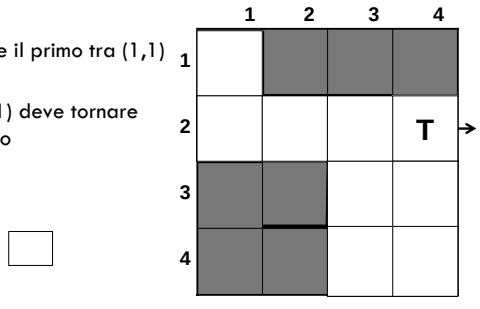


Ricerca in profondità *online*

- Gli agenti *online* ad ogni passo decidono l'azione da fare (non il piano) e la eseguono.
- Ricerca in profondità *online*
  - Un metodo locale
  - Il backtracking significa tornare sui propri passi
  - È necessario ricordarsi ciò che si è scoperto
  - Esplorazione sistematica delle alternative

Esempio

- Sceglie il primo tra (1,1) e(2,2)
- In (1, 1) deve tornare indietro



Algoritmo DF *online*

function Agente-Online-DFS(s') returns *un'azione*
  static: risultato, notexpl, notback,
    s (*stato precedente*), a (*ultima azione*)
if Goal-Test(s') **then return** stop
if s' *nuovo stato* **then** notexpl[s'] ← Azioni(s')
if s *non è* null **then** risultato[a, s] ← s'; notback[s'] ← s
if notexpl[s'] *vuoto* **then**
  if notbackl[s'] *vuoto* **then return** stop
  else a ← *azione per tornare in* POP(notback[s'])
else a ← POP(notexpl[s'])
s ← s'; return a

Ricerca euristica *online*

- Nella ricerca online si conosce il valore della f. euristica solo una volta esplorato lo stato.
- Un algoritmo di tipo Best First non funzionerebbe.
- Serve un metodo locale
- Random-restart non praticabile
- Come sfuggire a minimi locali?

Due soluzioni

- Random-walk
- Apprendimento Real-Time: esplorando si aggiustano i valori dell'euristica per renderli più realistici
  - Se s è uno stato con f. euristica h(s), i successori sono valutati h(s) se inesplorati, altrimenti si prende la loro stima h + costo azione per raggiungerli.
  - Costo-LRTA*(s, a, s', H) = h(s) *se s' non esplorato*
    H(s') + costo(s,a,s') *altrimenti*

LRTA*

function Agente-LRTA*(s') **returns** *un'azione*
static: risultato, H, s, a
  if Goal-Test(s') **then return** stop
  if s' nuovo **then** H[s'] ← h[s']
1. **if** s ≠ null
    risultato[a, s] ← s'
    H[s] ← *min* Costo-LRTA*(s, b, risultato[b, s], H)
2. a ← un'azione tale che
      min Costo-LRTA*(s', b, risultato[b, s'], H)
   s ← s'; return a

Esempio di Real Time Learning A*