Lecture 2

① Classical Planning . state Model S(P)

① state space S

② initial state $s_0 \in S$

③ ~~S_G~~ set $S_G \subseteq S$, goal states.

④ Actions $A(s) \subseteq A$

⑤ deterministic transition function $s' = f(a,s)$

⑥ action costs $c(a,s)$

Solutions: actions that map s_0 into S_G.

② Blind search

DFS. BFS. UCS. ID.

Heuristic search

A^*. WA^*. IDA^*. Hill climbing.

Best first search.

③ completeness: ~~go~~ guaranteed to find a solution

optimality: solution optimal ?

time complexity

Space complexity

Branching factor b 有几个分支

goal depth d 找几次

④ Breadth-first-search: 找邻接节点, 队列 queue FIFO

变形 uniform cost search: 每次找 $g(s)$ 最小的 ~~stack~~ bfs priority queue

depth-first search: 不撞南墙不回头 LIFO stack.

Iterative deepening: combine bfs and dfs stack.

limit = 0 BFS

limit = 1 BFS

limit = 2 BFS

⋮

⑤. Systematic Heuristic Search:

h^* perfect heuristic remaining cost

Greedy best-first Search ⟸ 用最小 h 进 priority queue

WA^* $g(s) + W * h(s)$.

A^*: $g(s) + h(s)$ ⟵ $h(s)$ 来选择 priority queue

IDA^*

g: 经过的路

h: 估算成本

Local Heuristic Search:

Hill-climbing

Enforced-hill-climbing

⑥ Safe: $h^*(s) = \infty \Rightarrow h(s) = \infty$ for $s \in S$

goal-aware: $h(s) = 0$ for goal states

admissible: 请 $h(s) \le h^*(s)$ for all s

consistent: $h(s) \le h(s') + c(a)$ for $s \xrightarrow{a} s'$

父节点 ~~的~~ h ~~≤~~ ≤ 子节点 h + cost.

指导 heuristic function 4个方法

consistent + goal-aware ⇒ admissible

admissible ⇒ goal-aware

admissible ⇒ safe

⑦ ~~Greedy best~~ hill-climbing

每次找局部最优 smaller h, 会局部停滞

Enforced hill-climbing only if $h(s) > 0$.

+ bfs + smaller h-value.

⑧

	DFS	BFS	ID	A^*	HC	IDA^*
complete	No	Yes	Yes	Yes	No	Yes
Optimal	No	Yes*	Yes	Yes	No	Yes
Time	∞	b^d	b^d	b^d	∞	b^d
Space	$b \cdot d$	b^d	$b \cdot d$	b^d	b	$b \cdot d$

A^*. IDA^* optimal if h is admissible

$h \le h^*$

BFS optimal if costs are same

$\boxed{\text{Lecture 3}}$

① $\boxed{\text{Classical Planning}}$

state space S

$\boxed{\text{initial state } s_0}$

S_G

actions $A(s)$

$\boxed{\text{deterministic } s' = f(a,s)}$

action cost $c(a,s)$

$\boxed{\text{Conformant planning}}$

state space S

$\boxed{\text{set of possible } s_0}$

S_G

actions $A(s)$

$\boxed{\text{non-deterministic } F(a,s)}$

action cost $c(a,s)$

MDPs

State Space S

s_0

S_G

$A(s)$

transition probabilities $P_a(s'|s)$

$C(a,s)$

POMDPs

states S

actions $A(s)$

transition prob $P_a(s'|s)$

initial belief state b_0

final belief state b_f

sensor model $P_a(o|s)$.

① actions - deterministic, + initial location known

$\Rightarrow$ classical

② action - stochastic + location observable

$\Rightarrow$ MDP

③ action stochastic + location partially observable

$\Rightarrow$ POMDP.

② $\boxed{\text{STRIPS}}$ $\Rightarrow P = \langle F, O, I, G \rangle$

$F \Rightarrow$ atoms (boolean)

$O \Rightarrow$ operators (actions) < add list $Add(o)$, delete list $del(o)$, precondition list $Prec(o)$

$I \Rightarrow$ initial situation

$G \Rightarrow$ goal situation

states $s \in S$ are collections of atoms $\boxed{F \cdot S = 2^F}$

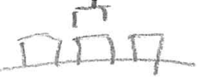
initial state s_0 is I

goal states $G \subseteq s$

actions a in O s.t. $prec(a) \subseteq s$

next state $s' = s - Del(a) + Add(a)$

action costs $C(a,s) = 1$

③ Example 

$\boxed{\text{Pre}}$: $on(x,y)$, $onTable(x)$, $clear(x)$

$holding(x)$, $armEmpty()$

$\boxed{\text{Actions}}$: $stack(x,y)$, $unstack(x,y)$

$putdown(x)$, $pickup(x)$

$stack(x,y)$: pre{$holding(x)$, $clear(y)$}

add($on(x,y)$, $clear(x)$, $armEmpty()$)

del{ ~~holding~~ $clear(y)$, $holding(x)$ }

$\boxed{\text{Lecture 12}}$ Reward shaping

$\boxed{\text{add additional rewards}}$

$$Q(s,a) = Q(s,a) + \alpha[r + F(s,s') + \gamma \max Q(s',a') - Q(s,a)]$$

$$\boxed{F(s,s') = \gamma\Phi(s') - \Phi(s)}$$

potential function

$$\Phi(s) \quad \frac{1}{|x(g) - x(s)| + |y(g) - y(s)|}$$

goal Manhattan distance

$\boxed{\text{n-step temporal difference learning}}$

$\boxed{\text{discounted future reward}}$

$$G(t) = r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n V(t+n)$$

n step

$$G_t^n = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots + v$$

$$\boxed{Q(s,a) = Q(s,a) + \alpha[G_t + \gamma^n Q(s',a') - Q(s,a)]}$$

$$\boxed{G_t = \sum_{i=t+1}^{t+n} \gamma^{i-t-1} r_i}$$

Lecture 4 Relaxation

① P, h^*

define P', h'^* used to estimate h^*

transformation, r $\Rightarrow$ $P \Rightarrow P'$

given $\Pi \in P$, estimate $h^*(\Pi)$ by $h'^*(r(\Pi))$

② Relaxation ☆

relaxation of h^* is $R = (P', r, h'^*)$

$r: P \to P'$ and $h'^*: P' \to R_0^+ \cup \{\infty\}$

$h^R(\Pi) = h'^*(r(\Pi)) \Leftarrow \boxed{h^R(\Pi) \le h^*(\Pi)}$ ☆

① native is $P' \subseteq P$ and $h'^* = h^*$

② efficiently constructible: $r(\Pi)$ 可构造

③ efficiently computable: $h'^*(\Pi')$ 可计算

③. $R = (P', r, h'^*)$: remove preconditions and deletes, h^*

Native? Yes.

eff. constructible: Yes.

eff. computable: NO, NP-hard

$\Rightarrow$ Approximate h^*.

Lecture 5 delete Relaxation Heuristic

① delete Relaxation: A^+ set of relaxed actions

STRIPS planning task $\Pi = (F, A, c, I, G)$

$\Pi^+ = (F, A^+, c, I, G)$

② dominance s'^+ dominates s^+ delete

③ $h^+(s)$ the cost of optimal relaxed plan for s

h^* : optimal plan. $h^* = 20$ $h^+ = 10$

用 h^{add} and h^{max} to estimate h^+.

$$h^{add}(s,g) = \begin{cases} 0 & g \subseteq s \\ \min_{a \in A, g \in add_a} c(a) + h^{add}(s, pre_a) & |g| = 1 \\ \sum_{g' \in g} h^{add}(s, \{g'\}) & |g| > 1 \end{cases}$$

$$h^{max}(s,g) = \begin{cases} 0 & g \subseteq s \\ \min_{a \in A, g \in add_a} c(a) + h^{max}(s, pre_a) & |g| = 1 \\ \max_{g' \in g} h^{max}(s, \{g'\}) & |g| > 1 \end{cases}$$

$h^{max} \le h^+ \Rightarrow h^{max} \le h^*$ optimistic

$h^{add} \ge h^+ \Rightarrow$ exist $h^{add}(s) > h^*(s)$

pessimistic

④ Best supporter from h^{add} / h^{max}.

$bs^{max}(p) = \arg\min c(a) + h^{max}(s, pre_a)$

$bs^{add}(p) = \arg\min c(a) + h^{add}(s, pre_a)$

得出的 action 画表. 得出最小 cost 的 action

$\Rightarrow$ $h^{FF}(s)$ 对于 action

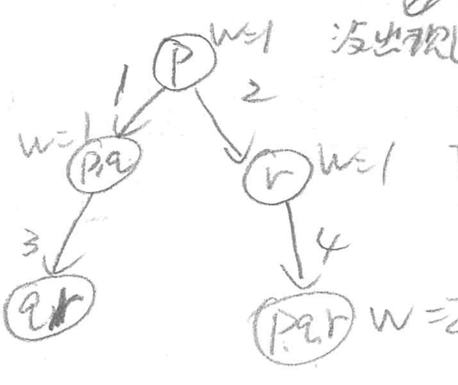
$h^{FF} \ge h^+ \Rightarrow$ exist $h^{FF}(s) > h^*(s)$

may be inadmissible

Lecture 6 Iterated width

① IW(k) BFS that prunes newly generated states whose novelty(s) > k

没出现过的最小长度



Novelty table

p	T
q	T
r	T
pq	T
pr	T
qr	T

Lecture 8 MDP

Bellman equation

$$\boxed{V(s)} = \max_{a \in A(s)} \sum_{s' \in S} P_a(s'|s)[r(s,a,s') + \gamma V(s')]$$

next state

$$Q(s,a) = \sum_{s' \in S} P_a(s'|s)[r(s,a,s') + \gamma V(s')]$$

action

$$\underline{V(s)} = \max_{a \in A(s)} Q(s,a)$$

最大Q值的action

Lecture 9 Value policy iteration

Value iteration

$$V(s) = \max_{a \in A} \sum_{s' \in S} P_a(s'|s)[r(s,a,s') + \gamma V(s')]$$

$$\boxed{V^{\pi}(s)} = \arg\max_{a \in A} Q^{\pi}(a,s)$$

action

$$Q^{\pi}(a,s) = \sum_{s' \in S} P_a(s'|s)[r(s,a,s') + \gamma V^{\pi}(s')]$$

$$V^{\pi}(s) = \sum_{s' \in S} P_a(s'|s)[r(s,a,s') + \gamma V^{\pi}(s')]$$

Lecture 10 Monte Carlo Tree Search

① Select ② expand ③ simulate ④ Back propagation

$$V(s) = \max \sum P_a(s'|s)[r(s,a,s') + \gamma V(s')]$$

UCB $\pi(s) = \arg\max Q(s,a) + \sqrt{\dfrac{2 \ln N(s)}{N(s,a)}}$

UCT $\pi(s) = \arg\max Q(s,a) + 2 \times C_p \times \sqrt{\dfrac{2 \ln N(s)}{N(s,a)}}$

Lecture 11

Reinforcement learning

Q-learning

SARSA

Q learning: off policy

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \cdot \max_{a'} Q(s',a') - Q(s,a)]$$

SARSA on-policy

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \cdot Q(s',a') - Q(s,a)]$$

选定的

Lecture 12

Approximate Q-function

$$f(s,a) = \begin{Bmatrix} f_1(s,a) \\ f_2(s,a) \\ \vdots \\ f_n(s,a) \end{Bmatrix}$$ feature vector

weight vector:

$$w_i^a \leftarrow w_i^a + \alpha[r + \gamma \max Q(s',a') - Q(s,a)] \cdot f_i(s,a)$$

Q-learning

$$Q(s,a) = f_1(s,a) \cdot w_1^a + f_2(s,a) \cdot w_2^a + \dots = \sum_{i=0}^{n} f_i(s,a) \cdot w_i^a$$

Initial : on(A,C) , oT(C) , oT(B) , cl(A) , cl(B)

goal : { on(A,B) , on(B,C) }

h^{add}

id	cl(A)	cl(B)	cl(C)	~~cl(A)~~ oT(A)	oT(B)	oT(C)	on(A,C)	On(A,B)	on(B,C)	h(A)	h(B)	h(C)	armFree
0	0	0	∞	∞	0	0	0	∞	∞	∞	∞	∞	0
1	0	0	1	∞	0	0	0	∞	∞	1	1	∞	0
2	0	0	1	2	0	0	0	2	3 / 2	1	1	2	0

add / max

$\boxed{cl(C)}$ → unstack(A,C) cost = 1
↳ pre : { on(A,C), cl(A), uF } ⇒ 1.
0 0 0

$\boxed{oT A}$ → putdown(A) cost = 1
↳ pre : h(A) ⇒ ∞ ⇒ 2
∞

$\boxed{on(A,B)}$ → stack(A,B) cost = 1
↳ pre : { cl(B), h(A) ⇒ ∞ ⇒ 2.
0 ∞

$\boxed{on(B,C)}$ → stack(B,C) cost = 1
↳ pre : { cl(C), h(B) ⇒ ∞ ⇒ { 3 : add, 2 : max
~~∞~~ ∞

$\boxed{h(A)}$ → unstack(A,C) cost = 1
↳ pre : { cl(A), uF, on(A,C) } ⇒ 1.
0 0 0

$\boxed{h(B)}$ → ~~unstack~~ pickup(B) cost 1
↳ pre : { uF, cl(B), onT(B) } ⇒ 1.
0 0 0

$\boxed{h(C)}$ → ~~unstack~~(C) cost = 1
pickup ↳ pre { cl(C), uF, onT(C) } ⇒ ∞
∞ 0 0

$h^{add} = 5$

$h^{max} = 2$

bs(on(A,B)) = $\boxed{stack(A,B)}$ ☆.
↳ pre : { cl(B), h(A) }

bs(on(B,C)) = $\boxed{stack(B,C)}$.
↳ pre : { cl(C), h(B) }

bs(h(A)) = $\boxed{unstack(A,C)}$
↳ pre : { on(A,C), cl(A), uF }

bs(cl(C)) = ~~pre~~ $\boxed{unstack(A,C)}$ ↳ pre supported

bs(h(B)) = $\boxed{pickup(B)}$
↳ pre { uF, cl(B), oT(B) }

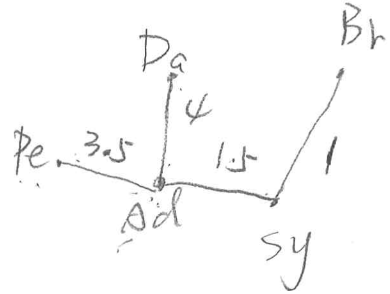
relaxed plan { unstack(A,C), pickup(B), stack(A,B), stack(B,C) }

$\boxed{h_{FF} = 4}$

☆ h^{max} h^{add} doesn't change best supported bs

i	at(Sy)	at(Ad)	at(Br)	at(Pe)	at(Da)	v(Sy)	v(Ad)	v(Br)	v(Pe)	v(Da)
0	0	∞	∞	∞	∞	0	∞	∞	∞	∞
1	0	1.5	1	~~∞~~ 5	~~∞ 5.5~~	0	1.5	1	∞	∞
2	0	1.5	1	5	5.5	0	1.5	1	5	~~∞~~ 5.5
3	0	1.5	1	5	5.5	0	1.5	1	5	5.5

I: at(Sy), v(Sy)
G: at(Sy), v(x) for all x



$$h^{add}(s,g) = \begin{cases} 0, & g \subseteq s \\ \min_{a \in A,\, g \in add_a} c(a) + h^{add}(s, pre_a), & |g| = 1 \\ \sum_{g' \in g} h^{add}(s, \{g'\}), & |g| > 1 \end{cases}$$

For i=1, at(Sy)

min(0, drive(Br,Sy) pre: at Br (1 + ∞), drive(Ad,Sy) 1 + (at Ad ∞))

at Br 1 + 0

at(Ad) drive(Sy, Ad)

min(1.5+0, 4+∞, 3.5+∞)

i	t(A)	t(B)	t(C)	t(D)	P(T)	P(A)	P(B)	P(C)	P(D)
0	0	∞	∞	∞	∞	∞	∞	0	∞
1	0	1	2	3	3	4	4.5	0	7
2	0	1	∞	∞	∞	∞	∞	0	∞
3	0	1	2	3	3	4	5/4	0	7/4
4									
5									

h^{add}

I: t(A) P(C)

Goal P(D) t

t(B) = 1 + 0

dr(A,B) ↳pre: {at A}

(C,B)

at B. min(∞, d(A,B), d(C,B)) 1+0, 1+∞

P(T) min load C 1 + 2 + 0 ↳pre: {at(C), P(C)}

P(A) unload(A) 1 + 0 + 3 ↳pre: {t(A), P(T)}

min{∞, 1 + h(P(T), t(A))} 1 + 2 + 0

at P(T) min[∞, 1 + h(at(C), P(C)) 1 + 2 + 0

P(B) unload(B) 1 + 1 + 3 ↳pre: {t(B), P(T)}

$h^{add} = 7$ load(C), l(C), l(B), l(D)

$h^{max} = 4$

P(D) 1 + 3 or 3 = ↳pre: {t(D), P(T)}