COMP90054 – Week 4 tutorial

Prepared by Zijie Xu. Last updated: 16 August 2023

Local search algorithms

Hill-climbing

- Always explores one successor node with its state minimising h
- Requires $h(s) > 0$ for $s \notin S^G$ to work
- Neither complete nor optimal
  - Might get stuck in local optima (cf. gradient descent)

Enforced Hill-climbing

- Use Breadth-First Search to look for a state with strictly smaller h-value
- In general, neither complete nor optimal
  - Complete if h is goal aware

Planning in AI

- A.k.a Model-based approach (cf. program-based, learning-based)
- We specify the model for a problem, a solver compute the controller automatically

State model

S State space S, finite and discrete
s_0 A **known** Initial state
S_G A set of goal states $S_G \subseteq S$
A A set of actions, with $A(s) \subseteq A$ for each $s \in S$
f A **deterministic** transition function $f: (a, s) \rightarrow s'$ for $a \in A(s)$
c **Positive action cost** functions $c(a, s)$

- A solution is an action sequence that maps s_0 into S_G
- A solution is optimal if it minimises sum of action costs

Conformant planning

S State space S, finite and discrete
S_0 A **set of possible initial states**
S_G A set of goal states $S_G \subseteq S$
A A set of actions, with $A(s) \subseteq A$ for each $s \in S$
F A **non-deterministic** transition function
c **Uniform** action cost functions $c(a, s)$

- A solution is an action sequence that must achieve the goal for any possible initial state and transition

MDPs – Markov Decision Processes

S State space S, finite and discrete
s_0 A **known** Initial state
S_G A set of goal states $S_G \subseteq S$
A A set of actions, with $A(s) \subseteq A$ for each $s \in S$
$P_a(s'|s)$ **Transition probabilities** $P_a(s'|s)$ for $s \in S$ and $a \in A(s)$
c **Positive action cost** functions $c(a, s)$

- A solution is a function/policy that maps states into actions
- A solution is optimal if it minimises expected cost to goal

POMDPs – Partially Observable Markov Decision Processes

S State space S, finite and discrete
b_0 An **initial belief state**
b_f A **final belief state**
A A set of actions, with $A(s) \subseteq A$ for each $s \in S$
$P_a(s'|s)$ **Transition probabilities** $P_a(s'|s)$ for $s \in S$ and $a \in A(s)$
$P_a(o|s)$ **Sensor model** that provides an observation to update belief states

- Belief states are probability distribution over S
- A solution is a function/policy that maps states into actions

STRIPS and PDDL

- Planning languages can be used to describe models
- **ST**anford **R**esearch **I**nstitute **P**roblem **S**olver
- **P**lanning **D**omain **D**efinition **L**anguage
- STRIPS ⊂ PDDL

A problem in STRIPS is a tuple $P = \langle F, O, I, G \rangle$

F set of all atoms (Boolean variables)
O set of all operators (3 elements: *Pre*, *Add*, *Del*)
I Initial situation $I \subseteq F$
G Goal situation $G \subseteq F$

- Core idea: use set of Boolean facts/atoms to represent system states
    - We add true atoms into set and remove false atoms from set
- We can use STRIPS to determine a state model
- Requirements:
    - F must contain all possible atoms
    - Negation are not allowed in G or *pre*