Artificial Intelligence

Blai Bonet

Universidad Simón Bolívar, Caracas, Venezuela

Non-deterministic state model

Model for non-deterministic problems

State models with non-deterministic actions correspond to tuples $(S, A, s_{init}, S_G, F, c)$ where

– S is finite set of states

– A is finite set of actions where $A(s)$, for $s \in S$, is the set of applicable actions at state s

– $s_{init} \in S$ is initial state

– $S_G \subseteq S$ is set of goal states

– $F : S \times A \to 2^S \setminus \{\emptyset\}$ is **non-deterministic** transition function. For $s \in S$ and $a \in A(s)$, $F(s, a) \neq \emptyset$ is set of **possible successors**

– $c : S \times A \to \mathbb{R}^{\geq 0}$ are **non-negative** action costs

Solutions

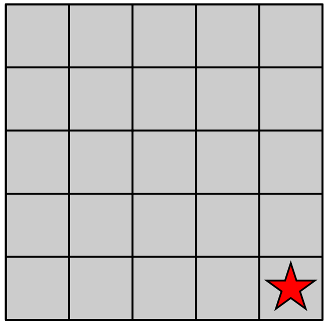
Solutions cannot be linear sequences of actions that map s_{init} into a goal state because actions are **non-deterministic**

Solutions are **strategies** or **policies** that map s_{init} into a goal state

Policies are functions $\pi : S \to A$ that map states s into actions $\pi(s)$ to apply at s (**implied constraint:** $\pi(s) \in A(s)$)

Unlike acyclic solutions for AND/OR graphs, solutions may be **cyclic**

Example



Goal: reach lower right corner $(n-1,0)$

Actions: Right (R), Down (D), Right-Down (RD)

Dynamics: R and D are deterministic, RD is non-deterministic.
If RD is applied at (x,y), the possible effects are

$$(x+1 \bmod n, y), \quad (x+1 \bmod n, y-1 \bmod n), \quad (x, y-1 \bmod n)$$

Executions

Given state s, a **finite execution** starting at s is interleaved sequence of states and actions $\langle s_0, a_0, s_1, a_1, \ldots, a_{n-1}, s_n \rangle$ such that:

- $s_0 = s$
- $s_{i+1} \in F(s_i, a_i)$ for $i = 0, 1, \ldots, n-1$
- $s_i \notin S_G$ for $i < n$ (i.e. once goal is reached, execution ends)

Maximal execution starting at s is (possibly infinite) sequence $\tau = \langle s_0, a_0, s_1, a_1, \ldots \rangle$ such that:

- if τ is **finite**, it is a finite execution ending in a goal state
- if τ is **infinite**, each finite prefix of τ is a finite execution

Execution $(s_0, a_0, s_1, \ldots)$ is **execution for** π iff $a_i = \pi(s_i)$ for $i \geq 0$

Fairness

Let $\tau = \langle s_0, a_0, s_1, \ldots \rangle$ be a maximal execution starting at $s_0 = s$

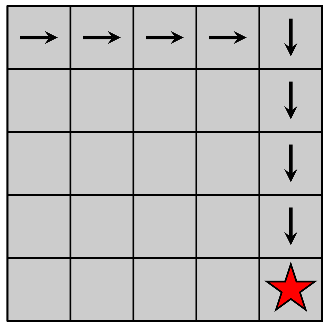
We say that τ is **fair execution** if either

- τ is finite, or
- if the pair (s,a) appears **infinitely often** in τ, then (s,a,s') also appears infinitely often in τ for every $s' \in F(s,a)$

Alternatively, τ is **unfair execution** iff

- τ is infinite, and
- there are s, a and $s' \in F(s,a)$ such that (s,a) appears i.o. in τ but (s,a,s') only appears a finite number of times

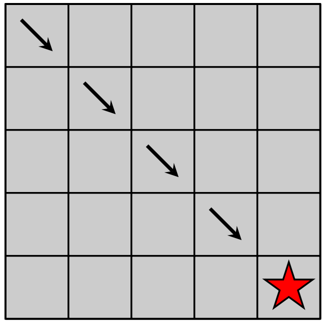
Example of fair and unfair executions



Policy π_1 given by

$$\pi_1(x,y) = \begin{cases} \text{Right} & \text{if } x < n-1 \\ \text{Down} & \text{if } x = n-1 \end{cases}$$

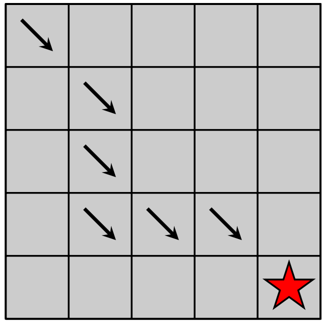
Example of fair and unfair executions



Policy π_2 given by

$$\pi_2(x, y) \;=\; \text{Right-Down}$$

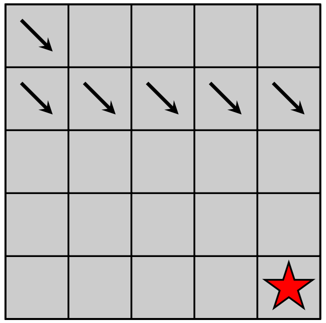
Example of fair and unfair executions



Policy π_2 given by

$$\pi_2(x, y) \;=\; \text{Right-Down}$$

Example of fair and unfair executions



Policy π_2 given by

$$\pi_2(x, y) \;=\; \text{Right-Down}$$

Solution concepts

Let $P = (S, A, s_{init}, S_G, F, c)$ be non-det. model and $\pi : S \to A$

– π is a **strong solution** for state s iff all **maximal executions** for π starting at s are finite (and thus, by definition, end in a goal state)

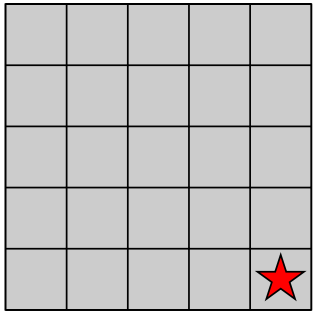
– π is a **strong cyclic solution** for state s iff every maximal execution for π starting at s that does not end in a goal state is **unfair**

The following can be shown:

– if π is a **strong solution** for state s, there is bound M such that all maximal executions have length $\leq M$

– if π is a **strong cyclic solution** for state s but not a strong solution, the length of the executions ending at goal states is unbounded

Solutions for example



- $\pi_1(x, y)$ given by Right when $x < n-1$ and Down when $x = n-1$ is a **strong solution**
- $\pi_2(x, y) =$ Right-Down is a **strong cyclic solution**
- $\pi_3(x, y) =$ Down is not a solution

Assigning costs to policies

If π is a strong solution for s_{init}, all maximal executions are bounded in length and a cost can be assigned

If π is a strong cyclic solution for s_{init} but not a strong solution, it is **not clear** how to assign cost to π

However, we can consider **best** and **worst** costs for π at s:

$$V^{\pi}_{best}(s) = \begin{cases} 0 & \text{if } s \in S_G \\ c(s, \pi(s)) + \min\{V^{\pi}_{best}(s') : s' \in F(s, \pi(s))\} & \text{if } s \notin S_G \end{cases}$$

$$V^{\pi}_{worst}(s) = \begin{cases} 0 & \text{if } s \in S_G \\ c(s, \pi(s)) + \max\{V^{\pi}_{worst}(s') : s' \in F(s, \pi(s))\} & \text{if } s \notin S_G \end{cases}$$

Characterizing solution concepts

Characterization of solutions using best/worst costs for π at s:

- π is **strong solution** for s_{init} iff $V^{\pi}_{worst}(s_{init}) < \infty$
- π is **strong cyclic solution** for s_{init} iff $V^{\pi}_{best}(s) < \infty$ for every state s that is **reachable** from s_{init} using π

Additionally, if we define $V_{\max}(s)$ at each state s by

$$V_{\max}(s) = \begin{cases} 0 & \text{if } s \in S_G \\ \min\limits_{a \in A(s)} \Big\{ c(s, a) + \max\{V_{\max}(s') : s' \in F(s, a)\} \Big\} & \text{if } s \notin S_G \end{cases}$$

Then, there is strong policy starting at s iff $V_{\max}(s) < \infty$

AND/OR formulation for non-deterministic models

Given non-deterministic model $P = (S, A, s_{init}, S_G, F, c)$, we define two different AND/OR graphs:

- **(Finite) AND/OR graph over states:** "OR vertices" for states $s \in S$ and "AND vertices" for pairs $\langle s, a \rangle$ for $s \in S$ and $a \in A(s)$. Connectors $(s, \{\langle s, a \rangle\})$ with cost $c(s, a)$ for $a \in A(s)$, and connectors $(\langle s, a \rangle, F(s, a))$ with zero cost
- **(Infinite but acyclic) AND/OR graph over executions:** "OR vertices" are finite executions starting at s_{init} and ending in states, and "AND vertices" are finite executions starting at s_{init} and ending in actions. Connectors $(\langle s_{init}, \ldots, s \rangle, \{\langle s_{init}, \ldots, s, a \rangle\})$ with cost $c(s, a)$ for $a \in A(s)$, and connectors $(\langle s_{init}, \ldots, s, a \rangle, \{\langle s_{init}, \ldots, s, a, s' \rangle : s' \in F(s, a)\})$ with zero cost

Finding strong solutions

If there is a strong solution (i.e. if $V_{\max}(s_{init}) < \infty$), AO* finds one such solution when applied on the infinite and acyclic AND/OR graph

Conversely, if AO* solves the infinite and acyclic AND/OR graph, the solution found by AO* is a strong solution

Another method is to **select actions greedily with respect to $V_{\max}$**

Indeed, the policy π defined by

$$\pi(s) = \operatorname*{argmin}_{a\in A(s)} \Big\{ c(s,a) + \max\{V_{\max}(s') : s' \in F(s,a)\} \Big\}$$

is a strong solution starting at s_{init} when $V_{\max}(s_{init}) < \infty$

Finding strong cyclic solutions

These solutions are more complex than strong solutions (and so, the algorithms are not straightforward)

Let's begin by definining $V_{\min}$ in a similar way to $V_{\max}$:

$$V_{\min}(s) = \begin{cases} 0 & \text{if } s \in S_G \\ \min_{a\in A(s)} \Big\{ c(s,a) + \min\{V_{\min}(s') : s' \in F(s,a)\} \Big\} & \text{if } s \notin S_G \end{cases}$$

For every policy π and state s, we have

$$0 \;\leq\; V_{\min}(s) \;\leq\; V^{\pi}_{best}(s) \;\leq\; V^{\pi}_{worst}(s) \;\leq\; V_{\max}(s) \;\leq\; \infty$$

Finding strong cyclic solutions

Consider non-deterministic model $(S, A, s_{init}, S_G, F, c)$

Algorithm for finding a strong cyclic solution for s_{init}:

1. Start with $S' = S$ and $A' = A$
2. Find $V_{\min}$ for states in S' and actions in A' by **solving the equations** for $V_{\min}$
3. Remove from S' all states s such that $V_{\min}(s) = \infty$, and remove from $A'(s)$, for all states s, the actions leading to removed states
4. Iterate steps 2–3 until reaching a **fix point** (# iterations bounded by $|S|$)
5. If state s_{init} is removed, then there is **no strong cyclic solution** for s_{init}
6. Else, the greedy policy with respect to last $V_{\min}$, given by

$$\pi(s) \;=\; \operatorname*{argmin}_{a\in A'(s)} \Big\{ c(s,a) + \min\{V_{\min}(s') : s' \in F(s,a)\} \Big\},$$

is strong cyclic solution for s_{init}

Solving equations for $V_{\min}$ (and $V_{\max}$)

Idea is to use the equation defining $V_{\min}$ as **assignment** inside an iterative algorithm that **updates** initial iterate until **convergence**

Algorithm for finding $V_{\min}$ over states in S' and actions in A':

1. Start with initial iterate $V(s) = \infty$ for every $s \in S'$
2. Update V at each state $s \in S'$ as:

$$V(s) := \begin{cases} 0 & \text{if } s \in S_G \\ \min_{a\in A'(s)} \Big\{ c(s,a) + \min\{V(s') : s' \in F(s,a)\} \Big\} & \text{if } s \notin S_G \end{cases}$$

3. Repeat step 2 until reaching a **fix point** (termination is guaranteed!)

At termination, the last iterate V satisfies $V = V_{\min}$ (for $V_{\max}$ replace innermost $\min$ by $\max$, and start with iterate $V(s) = 0$ for all s)

Summary

- Non-deterministic state model
- Solution concepts: strong and strong cyclic policies
- Algorithms for computing solutions