Homework assignment MPRI 2024

Deadline: 19 November 2024 AOE

Notations

Let us first define stochastic arenas.

Definition 1 (Stochastic arenas)**.** *A stochastic arena is* $\mathcal{A} = (G, V_{\text{Max}}, V_{\text{Min}}, V_{\text{Random}}, \delta)$ *where*

- $G = (V, E)$ *is a graph and* $V = V_{\text{Max}} \uplus V_{\text{Min}} \uplus V_{\text{Random}}$ *partitions the vertices into those controlled by Max, Min, and random vertices.*
- $\delta : V_{\text{Random}} \to \mathcal{D}(E)$ *is the probabilistic transition function.*

Definition 2 (Stochastic reachability games)**.** *Let* $\mathcal{A}$ *a stochastic arena. A stochastic reachability game is* $\mathcal{G} = (\mathcal{A}, \texttt{Reach}(\textit{Win}))$ *with* $\textit{Win} \subseteq V$*.*

For simplicity we assume that Win is a sink (meaning a single vertex with a self-loop).

A *strategy* for Max is a function $\sigma : \text{Paths} \to \mathcal{D}(E)$, and similarly for Min. Note that a strategy is allowed to randomise over its actions. A pure strategy does not use randomisation: $\sigma : \text{Paths} \to E$, and a positional strategy does not use memory: $\sigma : V_{\text{Max}} \to \mathcal{D}(E)$.

When a pair of strategies (σ, τ) and an initial vertex u is fixed, we obtain a stochastic process: we write $\mathbb{P}^u_{\sigma,\tau}$ for the probability measure on infinite plays. We write $\mathbb{P}^u_{\sigma,\tau}(\texttt{Reach}(\text{Win}))$ for the probability that the infinite path reaches Win when Max plays σ, Min plays τ, and we start from u.

We say that the game is *determined* if $\text{val}^{\mathcal{G}}_{\text{Max}}(u) = \text{val}^{\mathcal{G}}_{\text{Min}}(u)$, and in that case define the *value* of u in $\mathcal{G}$ as $\text{val}^{\mathcal{G}}(u)$. A strategy σ is optimal from u if $\text{val}^{\sigma}(u) = \text{val}^{\mathcal{G}}(u)$, and simply optimal if it is optimal from all vertices.

Reminders

Theorem 1 (Pure positional determinacy for stochastic reachability games)**.** *Stochastic reachability games are uniformly purely positionally determined.*

Let us consider a stochastic reachability game $\mathcal{G}$. Let Y the set of functions $\mu : V \to [0, 1]$, it is a lattice when equipped with the componentwise order. We define the operator $\mathbb{O}^{\mathcal{G}} : Y \to Y$ by:

$$\mathbb{O}^{\mathcal{G}}(\mu)(u) = \begin{cases} 1 & \text{if } u \in \text{Win}, \\ \max\{\mu(v) : u \to v \in E\} & \text{if } u \in V_{\text{Max}}, \\ \min\{\mu(v) : u \to v \in E\} & \text{if } u \in V_{\text{Min}}, \\ \sum_{v \in V} \delta(u)(v) \cdot \mu(v) & \text{if } u \in V_{\text{Random}}. \end{cases}$$

Since $\mathbb{O}^{\mathcal{G}}$ is monotonic, it has a least fixed point, which is also the least pre-fixed point.

Theorem 2. *Let* $\mathcal{G}$ *a stochastic reachability game. The least fixed point of* $\mathbb{O}^{\mathcal{G}}$ *computes the values of* $\mathcal{G}$*. Furthermore, any uniform pure positional strategy* τ *for Min that satisfies*

$$u \in V_{\text{Min}} : \ \tau(u) \in \textit{argmin}\left\{\textit{val}^{\mathcal{G}}(v) : u \to v \in E\right\}$$

is optimal.

Given a strategy σ, we write $\mathcal{G}[\sigma]$ for the game obtained by restricted the moves of Max to those prescribed by σ. It is a one-player game, since only Min has choices to make.

The value iteration algorithm

Algorithm 1: The value iteration algorithm.

Data: A stochastic reachability game.
Choose μ
while *true* **do**
  $\mu \leftarrow \mathbb{O}^{\mathcal{G}}(\mu)$
return μ

Question 1. *Prove that if $\mu \leqslant val^{\mathcal{G}}$ and $\mu \leqslant \mathbb{O}^{\mathcal{G}}(\mu)$, then the value iteration computes in the limit the values. Suggest two different ways of choosing μ satisfying these assumptions. Explain why the statement in the previous version of this homework was not easy to prove: it is not clear why stopping when $||\mathbb{O}^{\mathcal{G}}(\mu) - \mu|| \leqslant \varepsilon$ would yield an ϵ-approximation of the values (it actually is still an open problem, which is only known to hold when the operator $\mathbb{O}$ is contracting).*

The strategy improvement algorithm

The key idea behind strategy improvement is to use val^σ to improve the strategy σ by *switching edges*, which is an operation that creates a new strategy. This involves defining the notion of *improving edges*: let us consider a vertex $u \in V_{\text{Max}}$, we say that $e : u \to v$ is an improving edge if

$$\text{val}^\sigma(v) > \text{val}^\sigma(u).$$

Intuitively: according to val^σ, playing e is better than playing $\sigma(u)$.

Given a strategy σ and a set of improving edges S (for each $u \in V_{\text{Max}}$, S contains at most one outgoing edge of u), we write $\sigma[S]$ for the strategy

$$\sigma[S](u) = \begin{cases} e & \text{if there exists } e = u \to v \in S, \\ \sigma(v) & \text{otherwise.} \end{cases}$$

The difficulty is that an edge being improving does not mean that it is a better move than the current one in any context, but only according to the value function val^σ, so it is not clear that $\sigma[S]$ is better than σ. Strategy improvement algorithms depend on the following two principles:

- **Progress**: updating a strategy using improving edges is a strict improvement,
- **Optimality**: a strategy which does not have any improving edges is optimal.

The pseudocode of the algorithm is given in Algorithm 2. The algorithm is non-deterministic, in the sense that both the initial strategy and at each iteration, the choice of improving edge can be chosen arbitrarily. A typical choice, called the "greedy all-switches" rule, choosing for each $u \in V_{\text{Max}}$ a maximal improving edge, meaning

$$\text{argmax}\left\{\text{val}^\sigma(v) : u \to v \in E\right\}.$$

Let us write $\sigma \leqslant \sigma'$ if for all vertices u we have $\text{val}^\sigma(u) \leqslant \text{val}^{\sigma'}(u)$, and $\sigma < \sigma'$ if additionally $\neg(\sigma' \leqslant \sigma)$. We make the following observation.

In this homework we will prove the optimality property, but not the progress property.

Theorem 3 (Progress property for the strategy improvement)**.** *Let σ a strategy and S a set of improving edges. We let $\sigma' = \sigma[S]$. Then $\sigma < \sigma'$.*

Let σ be any strategy of Max and S a set of improving edges. We let $\sigma' = \sigma[S]$.

Algorithm 2: The strategy improvement algorithm.

```
Choose an initial strategy σ_0 for Max
for i = 0, 1, 2, ... do
    Compute val^{σ_i} and the set of improving edges
    if σ_i does not have improving edges then
        return σ_i
    Choose a non-empty set S_i of improving edges
    σ_{i+1} ← σ_i[S_i]
```

Question 2. *Prove the following properties:*

- $\mathit{val}^\sigma \leqslant \mathbb{O}^{\mathcal{G}[\sigma']}(\mathit{val}^\sigma) \leqslant \mathbb{O}^{\mathcal{G}}(\mathit{val}^\sigma)$.
- *The inequality* $\mathit{val}^\sigma(u) \leqslant \mathbb{O}^{\mathcal{G}}(\mathit{val}^\sigma)(u)$ *is strict if and only if* u *is a vertex of Max that has at least one improving edge.*
- *The inequality* $\mathit{val}^\sigma(u) \leqslant \mathbb{O}^{\mathcal{G}[\sigma']}(\mathit{val}^\sigma)(u)$ *is strict if* u *has an outgoing edge in* S.
- *If* S *is constructed by the greedy all-switches rule, then* $\mathbb{O}^{\mathcal{G}[\sigma']}(\mathit{val}^\sigma) = \mathbb{O}^{\mathcal{G}}(\mathit{val}^\sigma)$.

Question 3 (Optimality property for the strategy improvement)**.** *Prove that if* σ *is a strategy that has no improving edges, then* σ *is optimal.*

Question 4. *Conclude that the strategy improvement algorithm is correct, independently of the choice of sets of improving edges.*

Question 5. *Let us write* $\sigma_0 < \sigma_1 < \sigma_2 < \cdots$ *for the sequence of positional strategies in an execution of the strategy improvement algorithm with the greedy all-switches rule over* $\mathcal{G}$ *and* $\mu_0 \leqslant \mu_1 \leqslant \mu_2 \leqslant \cdots$ *for the sequence of functions computed by the corresponding value iteration algorithm over* $\mathcal{G}$ *initialised at* val^{σ_0}*: for all* k*, we have* $\mu_0 = \mathit{val}^{\sigma_0}$ *and* $\mu_{k+1} = \mathbb{O}^{\mathcal{G}}(\mu_k)$.

Prove that for all k*, we have* $\mu_k \leqslant \mathit{val}^{\sigma_k}$*. What does this say about these two algorithms?*

We now look at some properties of the influence of the choice of the sets of improving edges for the strategy improvement algorithm.

Question 6. *Find an example where the greedy all-switches rule is suboptimal, meaning another sequence of choices of sets of improving edges yields less iterations than greedy all-switches.*

Question 7. *Prove that there exists a sequence of choices of sets of improving edges which makes strategy improvement terminate in at most* $|V_{\mathrm{Max}}|$ *many iterations.*