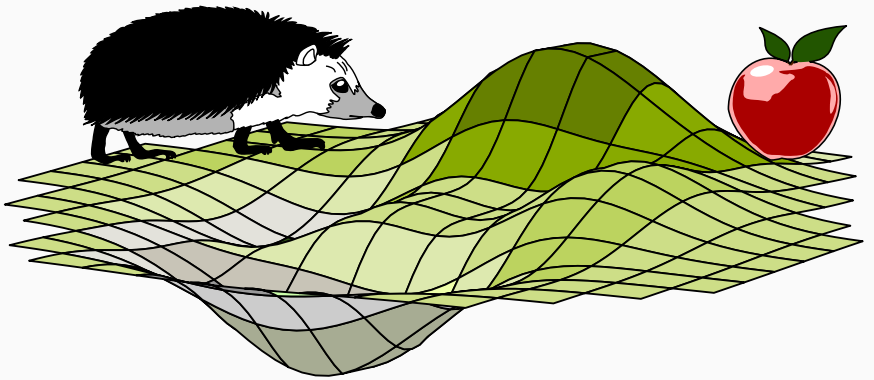
REINFORCEMENT LEARNING

DYNAMIC PROGRAMMING

Pavel Osinenko



Dynamic programming

Consider the following ∞-horizon opt. control problem for an MDP:

$$\max_{\kappa} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \rho(X_k, \kappa(X_k)) \mid X_0 = x\right]$$

Denote: $X_+^u \sim P_x(x \mid x, u)$, the next state

Claim:

$$V(x) = \max_u \left\{\rho(x,u) + \gamma\, \mathbb{E}\left[V(X_+^u)\right]\right\} \quad *$$

Let's check. First, recall:

$$V(x) = \max_{\kappa} \mathbb{E}\left[\sum_{k=0} \gamma^k \rho(X_k, \kappa(X_k)) \mid X_0 = x\right]$$

* Omitting the condition under $\mathbb{E}$ for brevity

Dynamic programming

(cont.)

Denote $J(x|\pi) := \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \rho(X_k, \pi(X_k)) \mid X_0 = x\right]$

The optimal policy

$$\pi^*(x) = \arg\max_{\pi} J(x|\pi)$$

Let's define $\hat{\pi}(u|\eta) := \{u, \eta\}$, concatenation of action u with some arbitrary tail policy η.

Dynamic programming

(cont.)

Let's unwrap the value function:

$$V(x) = \max_{\kappa} \left\{ \rho(x, \kappa(x)) + \gamma \, \mathbb{E}\left[J(X_+^{\kappa} \mid \kappa) \right] \right\}$$

$$= \max_{\hat{\kappa}} \left\{ \rho(x, u) + \gamma \, \mathbb{E}\left[J(X_+^{u} \mid \eta) \right] \right\}$$

By the virtue of V,

$$\max_{\hat{\kappa}} \left\{ \rho(x, u) + \gamma \, \mathbb{E}\left[J(X_+^{u} \mid \eta) \right] \right\} \leq$$

$$\max_{\hat{\kappa}} \left\{ \rho(x, u) + \gamma \, \mathbb{E}\left[V(X_+^{u}) \right] \right\} =$$

$$\max_{u} \left\{ \rho(x, u) + \gamma \, \mathbb{E}\left[V(X_+^{u}) \right] \right\}$$

Dynamic programming

(cont.)

On the other hand,

$$\max_{\hat{K}} \{ \rho(x,u) + \gamma \mathbb{E}[J(X_+^u \mid \eta)] \} \geq$$

$$\rho(x,u) + \gamma \mathbb{E}[J(X_+^u \mid \eta)], \quad \forall u, \eta$$

So, in particular,

$$\max_{\hat{K}} \{ \rho(x,u) + \gamma \mathbb{E}[J(X_+^u \mid \eta)] \} \geq$$

$$\rho(x, K^*(x)) + \gamma \mathbb{E}[J(X_+^{K^*(x)} \mid K^*)] = V(x)$$

Combining this fact with the conclusion of the previous slide, the original claim follows

Dynamic programming

This fact

$$V(x) = \max_u \{ \rho(x,u) + \gamma \mathbb{E}[V(X_+^u)] \}$$

is a particular expression of the dynamic programming principle (DPP) which, informally, states:

> the optimal tail policy is exactly the tail of the full optimal policy

Observe that the maximization in the above is over actions, not policies!

This is an extremely powerful fact characterizing the value function

Dynamic programming

Let's now work out the DPP in CT case.

We start with

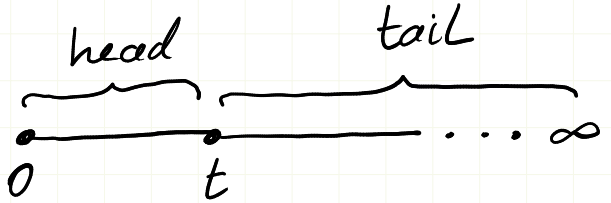
$$J(x|\kappa) = \mathbb{E}\left[\int_0^\infty e^{-\gamma\tau} \rho(X_\tau, \kappa(X_\tau)) d\tau \,\middle|\, X_0 = x \right]$$

Recall

$$V(x) = \max_\kappa J(x|\kappa)$$

$$\kappa^*(x) = \arg\max_\kappa J(x|\kappa)$$

Split the horizon as:



Dynamic programming

(cont.)

We have:

$$V(x) = \mathbb{E}\left[\int_0^t e^{-\gamma\tau}\rho\left(X_\tau^*, \kappa^*\left(X_\tau^*\right)\right)d\tau \mid X_0^* = x\right] +$$

$$\mathbb{E}\left[\mathbb{E}\left[\int_t^\infty e^{-\gamma\tau}\rho\left(X_\tau^*, \kappa^*\left(X_\tau^*\right)\right)d\tau \mid X_0^* = X_t^*\right]\right],$$

where X_t^* is the trajectory* under κ^*

* The trajectory stands for the stochastic process that is the strong solution to the respective initial value problem (IVP) with the system SDE, assuming such a solution exists

Dynamic programming

(cont.)

We have:

$$\mathbb{E}\left[\int_t^\infty e^{-\gamma\tau}\rho\left(X_\tau^*, \kappa^*\left(X_\tau^*\right)\right)d\tau \mid X_t^* = x_t\right] = \left(\text{pull out } e^{-\gamma t}\right)$$

$$e^{-\gamma t}\,\mathbb{E}\left[\int_t^\infty e^{-\gamma(\tau-t)}\rho\left(X_\tau^*, \kappa^*\left(X_\tau^*\right)\right)d\tau \mid X_t^* = x_t\right] = \left(\begin{array}{l}\text{subs.}\\ \theta := \tau - t\end{array}\right)$$

$$e^{-\gamma t}\,\mathbb{E}\left[\int_0^\infty e^{-\gamma\theta}\rho\left(X_{\theta+t}^*, \kappa^*\left(X_{\theta+t}^*\right)\right)d\theta \mid X_0^* = x_t\right] = \left(\begin{array}{l}\text{introduce}\\ \hat{X}_\theta^* := X_{\theta+t}^*\end{array}\right)$$

$$e^{-\gamma t}\,\mathbb{E}\left[\int^\infty e^{-\gamma\theta}\rho\left(\hat{X}_\theta^*, \kappa^*\left(\hat{X}_\theta^*\right)\right)d\theta \mid X^* = x_t\right] = \ ^*$$

$$e^{-\gamma t}\,v\left(x_t^*\right)$$

* X_t, $\hat{X}_\theta$ are time-homogeneous Markov processes, whose distributions are uniquely determined by the drift, diffusion and the optimal policy

Dynamic programming

(cont.)

Thus, we get:

$$V(x) = \mathbb{E}\left[\int_0^t e^{-\gamma\tau} \rho\left(X_\tau^*, \kappa^*\left(X_\tau^*\right)\right) d\tau \mid X_0^* = x\right] + e^{-\gamma t}\, \mathbb{E}\left[V\left(X_t^*\right)\right]$$

Let's introduce an arbitrary head policy $\kappa^{\#}_{[0,t]}$ and a concatenation as follows: $\kappa^{\#} = \left\{\kappa^{\#}_{[0,t]}, \kappa^{*}_{[t,\infty)}\right\}$

(cont.)

Then, by definition of V,

$$V(x) \geq J(x \mid \kappa^{\#}) =$$

$$\mathbb{E}\left[\int_0^t e^{-\gamma\tau} \rho\left(X_\tau^{\#}, \kappa^{\#}_{[0,t]}\left(X_\tau^{\#}\right)\right) d\tau \mid X_0^{\#} = x \right] +$$

$$\mathbb{E}\left[\mathbb{E}\left[\int_t^\infty e^{-\gamma\tau} \rho\left(X_\tau^{\#}, \kappa^{*}\left(X_\tau^{\#}\right)\right) d\tau \mid X_0^{\#} = X_t^{\#} \right] \right],$$

where $X_t^{\#}$ is the trajectory under $\kappa^{\#}$

Dynamic programming

(cont.)

Using the same head-tail splitting technique as before, we get

$$V(x) \geq \mathbb{E}\left[\int_0^t e^{-\gamma\tau}\rho\left(X_\tau^{\#}, K_{[0,t]}^{\#}\left(X_\tau^{\#}\right)\right)d\tau \mid X_0^{\#} = x\right] + e^{-\gamma t}\mathbb{E}\left[V\left(X_t^{\#}\right)\right]$$

$$= \mathbb{E}\left[\int_0^t e^{-\gamma\tau}\rho\left(X_\tau^{\#}, K_{[0,t]}^{\#}\left(X_\tau^{\#}\right)\right)d\tau + e^{-\gamma t}V\left(X_t^{\#}\right) \mid X_0^{\#} = x\right]$$

And this holds for any $K^{\#}$ defined as above, whence

$$V(x) \geq \max_{K_{[0,t]}^{\#}}\left\{\mathbb{E}\left[\int_0^t e^{-\gamma\tau}\rho\left(X_\tau^{\#}, K_{[0,t]}^{\#}\left(X_\tau^{\#}\right)\right)d\tau + e^{-\gamma t}V\left(X_t^{\#}\right) \mid X_0^{\#} = x\right]\right\}$$

Dynamic programming

(cont.)

But

$$\max_{\substack{\# \\ K_{[0,t]}}} \left\{ \mathbb{E}\left[\int_0^t e^{-\gamma\tau} \rho\left(X_\tau^{\#}, K_{[0,t]}^{\#}\left(X_\tau^{\#}\right)\right) d\tau + e^{-\gamma t} V\left(X_t^{\#}\right) \middle| X_0^{\#} = x \right] \right\}$$

has a fixed tail, whence, it effectively equals

$$\max_{K} \left\{ \mathbb{E}\left[\int_0^t e^{-\gamma\tau} \rho\left(X_\tau, K\left(X_\tau\right)\right) d\tau + e^{-\gamma t} V\left(X_t\right) \middle| X_0 = x \right] \right\}$$

On the other hand, by optimality,

$$\max_{\substack{\# \\ K_{[0,t]}}} \left\{ \mathbb{E}\left[\int_0^t e^{-\gamma\tau} \rho\left(X_\tau^{\#}, K_{[0,t]}^{\#}\left(X_\tau^{\#}\right)\right) d\tau + e^{-\gamma t} V\left(X_t^{\#}\right) \middle| X_0^{\#} = x \right] \right\} \geqslant$$

$$\mathbb{E}\left[\int_0^t e^{-\gamma\tau} \rho\left(X_\tau^{*}, K^{*}\left(X_\tau^{*}\right)\right) d\tau + e^{-\gamma t} V\left(X_t^{*}\right) \middle| X_0^{*} = x \right]$$

(we used K^* formally as a particular candidate)

Dynamic programming

(cont.)

But

$$\mathbb{E}\left[\int_0^t e^{-\gamma\tau}\rho\left(X_\tau^*, \kappa^*(X_\tau^*)\right)d\tau + e^{-\gamma t}V\left(X_t^*\right) \mid X_0^* = x\right] = V(x)$$

as we showed before.

Combining all facts, we get:

$$V(x) = \max_\kappa \left\{\mathbb{E}\left[\int_0^t e^{-\gamma\tau}\rho\left(X_\tau, \kappa(X_\tau)\right)d\tau + e^{-\gamma t}V(X_t) \mid X_0 = x\right]\right\}$$

which is the DPP in CT case

Dynamic programming

Now, let's recall that the generator of environment SDE

$$dX_t = f(X_t, U_t)dt + \sigma(X_t, U_t)\, dB_t$$

reads, for a $\mathcal{C}^2$ function φ,

$$\mathcal{A}^u \varphi = \mathcal{L}_{f(x,u)}\, \varphi(x) + \frac{1}{2}\sigma^2(x,u)\varphi''(x),$$

where $\mathcal{L}$ is the Lie derivative.

Recall the Ito rule:

$$\varphi(X_t) = \varphi(X_0) + \int_0^t \varphi'(X_\tau)dB_\tau + \frac{1}{2}\int_0^t \varphi''(X_\tau)d\tau,$$

where the 1st integral is an Ito integral

Dynamic programming

(cont.)

Assuming V is sufficiently smooth, write

$$e^{-\gamma t} V(X_t^*) = V(x) + \int_0^t e^{-\gamma \tau} \left(-\gamma V(X_\tau^*) + \mathcal{A}^{\pi^*(X_\tau^*)} V(X_\tau^*) \right) d\tau + \langle \text{a martingale} \rangle$$

upon applying the Ito rule.

Dynamic programming

(cont.)

Now, add $\int_0^t e^{-\gamma\tau} \rho(X_\tau^*, \kappa^*(X_\tau^*))\, d\tau$ on both sides and take the conditional expectation under $X_0^* = x$ to get

$$\mathbb{E}\left[\int_0^t e^{-\gamma\tau} \rho(X_\tau^*, \kappa^*(X_\tau^*))\, d\tau + e^{-\gamma t} V(X_t^*) \mid X_0^* = x\right] =$$

$$V(x) + \mathbb{E}\left[\int_0^t e^{-\gamma\tau}\left(-\gamma V(X_\tau^*) + \mathcal{A}^{\kappa^*(X_\tau^*)} V(X_\tau^*) + \rho(X_\tau^*, \kappa^*(X_\tau^*))\right) d\tau \mid X_0^* = x\right]$$

Apply the DPP to conclude:

$$\mathbb{E}\left[\int_0^t e^{-\gamma\tau}\left(-\gamma V(X_\tau^*) + \mathcal{A}^{\kappa^*(X_\tau^*)} V(X_\tau^*) + \rho(X_\tau^*, \kappa^*(X_\tau^*))\right) d\tau \mid X_0^* = x\right] = 0$$

(check that)

Dynamic programming

(cont.)

We have:

$$\mathbb{E}\left[\int_0^t e^{-\gamma\tau}\left(-\gamma V(X_\tau^*) + \mathcal{A}^{\pi^*(X_\tau^*)} V(X_\tau^*) + \rho(X_\tau^*, \pi^*(X_\tau^*))\right) d\tau \,\middle|\, X_0^* = x\right] = 0$$

Divide by t, take $\lim_{t \searrow 0}$, noticing that $X_t^* \to x$, to get*:

$$-\gamma V(x) + \mathcal{A}^{\pi^*(x)} V(x) + \rho(x, \pi^*(x)) = 0$$

This can be cast into

$$\max_{u \in \mathcal{U}} \left\{ -\gamma V(x) + \mathcal{A}^u V(x) + \rho(x, u) \right\} = 0$$

* Assuming we can interchange $\mathbb{E}$ and $\lim$, and various functions involved are smooth

This

$$\max_{u \in \mathcal{U}} \left\{ -\gamma V(x) + \mathcal{A}^u V(x) + \rho(x,u) \right\} = 0$$

is the celebrated Hamilton-Jacobi-Bellman (HJB) equation

Recall that in the deterministic case, $\mathcal{A}^u V(x)$ is just $\mathcal{L}_{f(x,u)} V(x)$, i.e., there is no 2-order derivative term $\frac{1}{2} \mathrm{tr}\{\sigma^T(x,u) \nabla^2 V(x) \sigma(x,u)\}$

Dynamic programming

In general, V being smooth is an extremely strong assumption.

There are two general ways of tackling this

verification

- suppose we found a solution to HJB
- show that it's the value function
- kind of reverse way

viscosity solutions

- for a PDE $\mathcal{D}(v) = 0$, w is a viscosity super/subsolution if $\forall$ smooth φ, $w - \varphi$ has a local min (resp., max) at x, and
$$\mathcal{D}(\varphi) \geq 0 \quad (\text{resp., } \leq 0)$$
- we utilize suff-ly smooth test functions

Dynamic programming

Viscosity solutions are a kind of weak solutions.

Under some suitable bounds, Lipschitz and continuity conditions on the system and the instant. objective, compactness of U, the value function is a viscosity solution of the HJB.

Dynamic programming

If the instant. objective is time-dependent, DPP might not hold.

Example*:

$$\min_{\{u_k\}} \sum_{k=0}^{2} \rho_k(u_k) + \sup_{0 \le k \le 3} x_k$$

s.t. $x_{k+1} = x_k + u_k, \; x_0 = 0$

$0 \le x_k \le h = \text{const}$

$u_k \in \{-h, 0, h\}$

$\rho_0(u_0) = -u_0,$

$\rho_1(u_1) = u_1,$

$\rho_2(u_2) = -\frac{u_2}{2}$

Policy $\{u_k\}$	Cost
$\{0,0,0\}$	0
$\{0,0,h\}$	$-\frac{h}{2}$
$\{0,h,0\}$	h
$\{0,h,-h\}$	$\frac{3}{2}h$
$\{h,0,-h\}$	$-\frac{h}{2}$
$\{h,0,0\}$	$-h$
$\{h,-h,0\}$	$-2h$
$\{h,-h,h\}$	$-\frac{5}{2}h$

* Jones, M., & Peet, M. M. (2017, December). Solving dynamic programming with supremum terms in the objective and application to optimal battery scheduling for electricity consumers subject to demand charges. In 2017 IEEE 56th Annual Conference on Decision and Control (CDC) (pp. 1323-1329). IEEE.

Dynamic programming

(cont.)

So, the optimal policy is $\{h, -h, h\}$.

Consider the tail problem at $k = 2$.

Admissible actions are h, 0.

It turns out that taking 0 is optimal, but it is not the tail of the optimal policy!

Dynamic programming

Now, consider the following ∞-horizon optimal control problem:

$$\max_{K} \sum_{k=0}^{\infty} \gamma^k \rho(x_k, K(x_k))$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k), \; u_k \in \mathcal{U}, \; x_k \in \mathbb{R}^n, \; \forall k$$

The HJB reads in this case:

$$V(x) = \max_{u \in \mathcal{U}} \{ \rho(x, u) + \gamma V(f(x, u)) \}, \; \forall x$$

Let's introduce the following **dynamic programming (DP) operator**:

$$\mathcal{T}[W](x) := \max_{u \in \mathcal{U}} \{ \rho(x, u) + \gamma W(f(x, u)) \}, \; \forall x$$

($[\cdot]$ is to stress that we apply $\mathcal{T}$ to a function)

Dynamic programming

(cont.)

We show that $\mathcal{T}$ is contractive for discounted problems ($\gamma < 1$).

First, observe that

$$\rho(x,u) + \gamma W_1(f(x,u)) \leq \rho(x,u) + \gamma W_2(f(x,u))$$

whenever $W_1(x) \leq W_2(x)$, $\forall x$.

Therefore,

$$\max_{u \in U} \{\rho(x,u) + \gamma W_1(f(x,u))\} \leq \max_{u \in U} \{\rho(x,u) + \gamma W_2(f(x,u))\}$$

So, $\mathcal{T}[W_1] \leq \mathcal{T}[W_2]$, the monotonicity condition, the 1st Blackwell's condition for contractive mapping

Dynamic programming

(cont.)

The 2nd Blackwell's condition requires that $\mathcal{I}$ be discounting, namely,

$$\mathcal{I}[W+a](x) = \max_{u \in U} \{ p(x,u) + \gamma W(f(x,u)) + \gamma a \} = \mathcal{I}[W] + \gamma a$$

for any $a \geq 0$, which is evident.

Thus, $\mathcal{I}$ is contractive with modulus γ and we can apply the Banach's fixed point theorem to conclude that

$$V_i := \mathcal{I}[V_{i-1}], \quad i \in \mathbb{N}_{+0}$$

converges to a unique limit.

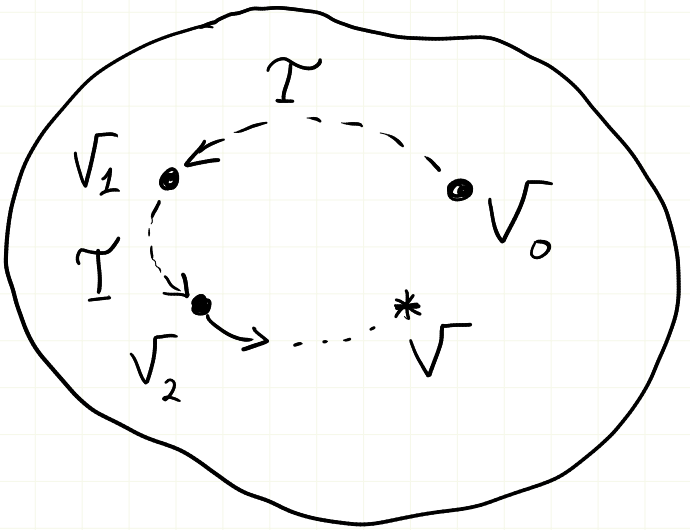
Dynamic programming

(cont.)

The rate of convergence reads:

$$\| T^i[V_0] - V \| \leq \frac{\gamma^i}{1-\gamma} \| T[V_0] - V_0 \|,$$

where T^i is the i-fold application of T, and $\| \bullet \|$ is the sup-norm.



Dynamic programming

Practical DP methods revolve around the HJB. Those are mostly tabular methods and seek to imitate the DP operator contraction.

What we just saw, namely, $V_i := \mathcal{T}[V_{i-1}]$ can be regarded as a kind of value update.

There is also policy update.

Combining these in different order gives value and policy iteration (VI, PI)

Dynamic programming

In the simplest form, i.e., for a DT environment, value iteration amounts to successive computations as per:

$$K_i(x) := \arg\max_{u \in \mathcal{U}} \left\{ \rho(x,u) + \gamma V_i(f(x,u)) \right\}, \forall x$$

$$V_{i+1}(x) := \rho(x, K_i(x)) + \gamma V_i(f(x, K_i(x))), \forall x$$

And now policy iteration:

$$V_i(x) := \rho(x, K_i(x)) + \gamma V_i(f(x, K_i(x))), \forall x$$

$$K_{i+1}(x) := \arg\max_{u \in \mathcal{U}} \left\{ \rho(x,u) + \gamma V_i(f(x,u)) \right\}, \forall x$$

Pay attention to the universal quantifiers!

Dynamic programming

The DPP can be formulated in different forms.

For instance, Watkins (1981) suggested a quality function, or Q-function:

$$Q(x,u) := \rho(x,u) + \gamma V(f(x,u))$$

The DPP now reads:

$$V(x) = \max_{u \in \mathcal{U}} Q(x,u)$$

The Q-function gives rise to Q-learning, which is a collection of DP & RL methods

Dynamic programming

Q-learning (QL) can be done in VI or PI forms, for instance, VI*:

$$K_i(x) := \arg\max_{u \in \mathcal{U}} Q_i(x,u), \quad \forall x$$

$$Q_{i+1}(x,u) := \rho(x,u) + \gamma Q_i(f(x,u), K_i(f(x,u))), \quad \forall x$$

* This is also known as SARSA, or on-policy QL

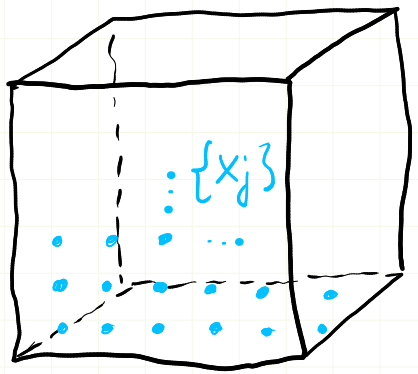
Dynamic programming

Recall the $\forall x$, the universal quantifier.

In practice, VI, PI, QL can rarely be implemented directly as they stand (and are thus not literally algorithms).

In DP, one takes a compact subset of the state space $\mathcal{X}$ and discretizes it.

Each step of DP thus amounts to updating tabular values of K_i, V_i or Q_i.



x_j	$K_i(x_j)$
⋮	⋮

x_j	$V_i(x_j)$
⋮	⋮

Dynamic programming

The size of the respective mesh in the state space grows rapidly with the dimension of the latter.
This is what is known as the curse of dimensionality

Therefore, DP is more suitable for environments with finite state and action spaces.

Dynamic programming

(cont.)

Example: frozen lake*

S	Start (Safe)
G	Goal (Safe, End)
F	Frozen (Safe)
H	Hole (Die, End)

S	F	F	H
F	F	F	F
F	H	F	F
F	F	H	G

Goal: to retrieve an item on a „lake" with dangerous spots

The state space is finite and small, so DP is suitable

* https://www.deeplearningwizard.com/