Q&A Prep: Conceptual Questions

Q&A Prep: Conceptual Questions

In this section I collect detailed answers to several conceptual questions that may come up during the presentation.

1. What does it mean to "solve" a min–max problem? Why is the solution at a saddle point?

The generic problem is

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} L(x,y),$$

where L is convex in x and concave in y.

Saddle point definition. A pair $(x^\star, y^\star)$ is a *saddle point* of L if

$$\forall x\in\mathcal{X},\ \forall y\in\mathcal{Y}:\qquad L(x^\star,y)\le L(x^\star,y^\star)\le L(x,y^\star).$$

Intuitively, $x^\star$ is the best response to $y^\star$ for the min-player, and $y^\star$ is the best response to $x^\star$ for the max-player. Neither player can unilaterally improve.

Connection to the minimax value. The *minimax value* is

$$V_{\mathrm{mm}} \quad := \quad \min_x \max_y L(x,y),$$

and the *maximin value* is

$$V_{\mathrm{mx}} \quad := \quad \max_y \min_x L(x,y).$$

In general we only have $V_{\mathrm{mm}}\ge V_{\mathrm{mx}}$. Under standard convex–concave and closedness assumptions, *strong duality* holds and

$$V_{\mathrm{mm}} = V_{\mathrm{mx}} = L(x^\star,y^\star),$$

for some saddle point $(x^\star,y^\star)$. In that case, solving the min–max problem is exactly the same as finding a saddle point.

Game-theoretic interpretation. We can interpret L as the payoff of a two-player zero-sum game: the min-player chooses x and wants to *decrease* L, the max-player chooses y and wants to *increase* L. A saddle point $(x^\star,y^\star)$ is precisely a Nash equilibrium of this zero-sum game: if the min-player deviates from $x^\star$ while the max-player keeps $y^\star$, the value can only go up; if the max-player deviates while the min-player keeps $x^\star$, the value can only go down.

How this maps to our project. In our convex MDP setting we consider

$$\min_{d \in \mathcal{D}} f(d),$$

where d is the occupancy measure and f is a convex functional. Taking the Fenchel dual of f gives

$$\min_{d \in \mathcal{D}} \max_{y \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}} \Big(\langle y, d \rangle - f^\star(y) \Big).$$

We then restrict d to be of the form d_π for some policy π and obtain the saddle-point problem

$$\min_\pi \max_y L(\pi, y) \; := \; \min_\pi \max_y \Big(\langle y, d_\pi \rangle - f^\star(y) \Big).$$

Here

- the **policy player** (min) chooses π,
- the **dual player** (max) chooses y.

A saddle point $(\pi^\star, y^\star)$ encodes:

- an optimal policy $\pi^\star$ for the original convex MDP,
- an optimal dual certificate $y^\star$.

So "solving the min–max problem" for us literally means "finding an optimal policy and its matching dual variable."

2. Why should alternating updates (first in x, then in y) make any sense? Gradient Descent–Ascent is not always convergent.

It is true that *naïve* Gradient Descent–Ascent (GDA)

$$x_{t+1} = x_t - \eta \nabla_x L(x_t, y_t), \qquad y_{t+1} = y_t + \eta \nabla_y L(x_t, y_t)$$

can fail even on simple bilinear problems: the iterates can spiral or cycle instead of converging. So we cannot simply assume that "updating one variable, then the other" will always work.

However, our algorithm is not arbitrary GDA: we use specific *online-learning / mirror-descent* style updates for each player that come with theoretical guarantees.

Online-learning view. Think of each iteration t as a round of a repeated game:

- the policy player picks π_t,
- the dual player picks y_t,
- they observe/estimate a loss $L(\pi_t, y_t)$ and gradients.

Each player runs an online learning algorithm (e.g., Follow-the-Leader for y, SPMA/mirror descent for π) whose performance is measured by *regret.* For the min-player,

$$R_{\min}(T) = \sum_{t=1}^{T} L(\pi_t, y_t) - \min_\pi \sum_{t=1}^{T} L(\pi, y_t),$$

and similarly for the max-player.

A classic result is: if both players have *sublinear regret*, $R_{\min}(T) = o(T)$ and $R_{\max}(T) = o(T)$, then the **averaged iterates**

$$\bar{\pi}_T := \frac{1}{T}\sum_{t=1}^{T} \pi_t, \qquad \bar{y}_T := \frac{1}{T}\sum_{t=1}^{T} y_t$$

converge to a saddle point in the sense that the *duality gap* $\mathrm{Gap}(\bar{\pi}_T, \bar{y}_T)$ goes to zero.

What we actually do. In our algorithm:

- The **dual player** y (or λ in CMDPs) is updated with a simple gradient / Follow-the-Leader step in a *convex* finite-dimensional space. Such algorithms are known to have low regret.
- The **policy player** uses SPMA, which is a mirror-descent method in the logit space with proven convergence rates in tabular MDPs.

We are therefore not using arbitrary GDA, but a specific pair of online algorithms that are chosen exactly because their interaction is known to converge (in the averaged sense) on convex–concave saddle-point problems.

In the presentation, I summarize this informally as:

> "Each player is using a low-regret online method (FTL for the dual, SPMA for the policy). Theory says that when both players have low regret, the average of their iterates converges to a saddle point of the min–max problem."

3. How do we think about convergence rates for min–max / saddle-point problems?

For a pure minimization problem we look at $f(x_t) - f(x^\star)$. For a saddle-point problem we use the *duality gap*.

Duality gap. Given any pair (x, y) define

$$\mathrm{Gap}(x, y) = \underbrace{\max_{y'} L(x, y')}_{\text{best response of max-player}} - \underbrace{\min_{x'} L(x', y)}_{\text{best response of min-player}}.$$

- At a saddle point $(x^\star, y^\star)$ we have $\mathrm{Gap}(x^\star, y^\star) = 0$.
- For arbitrary (x, y) the gap is nonnegative and measures how far we are from equilibrium.

Rate definition. A typical convergence statement looks like

$$\mathbb{E}\big[\mathrm{Gap}(\bar{x}_T, \bar{y}_T)\big] \le \frac{C}{T} \quad \text{or} \quad \le \frac{C}{\sqrt{T}},$$

for some constant C. This says the *averaged* iterates approach a saddle point at rate $O(1/T)$ or $O(1/\sqrt{T})$.

Connection to regret. If the min-player has regret $R_{\min}(T)$ and the max-player has regret $R_{\max}(T)$, one can show

$$\mathrm{Gap}(\bar{x}_T, \bar{y}_T) \;\le\; \frac{R_{\min}(T) + R_{\max}(T)}{T}.$$

So if each player has, say, $O(\sqrt{T})$ regret, we get an $O(1/\sqrt{T})$ bound on the duality gap, and so on.

How this applies to our project.

- For the *inner* SPMA oracle and the NPG–PD baseline, existing theory provides convergence rates in the tabular setting (e.g., linear convergence for SPMA).
- For the *full* Dual–SPMA algorithm with function approximation and approximate occupancy estimates, we do *not* derive a new formal rate; we treat understanding the overall rate as future work.

In the talk I phrase it as:

> "Formally, convergence is measured by the primal–dual gap. For the inner policy oracles, rates are known from the literature. For our full implementation with function approximation, we mainly check stability and empirical convergence; a precise rate is left as future work."

4. Why do we use RL at all? Why not just plug the min–max problem into a standard convex solver?

If we had a *small* tabular MDP with *known* transition probabilities, we could formulate the convex MDP entirely in terms of the occupancy measure d and solve

$$\min_{d \in \mathcal{D}} f(d)$$

directly with a generic convex optimization or LP solver.

Here:

- $\mathcal{D}$ is the set of all valid discounted occupancy measures (a convex polytope defined by linear flow constraints),
- f is a convex functional of d (e.g., negative reward plus entropy penalty or constraint penalties).

This is the "hidden convexity" of MDPs.

Why this is not enough for our setting. In realistic RL problems:

- we do *not* know the transition kernel $P(s' \mid s, a)$,
- the state space can be large or continuous,
- we only interact with the environment by *running* a policy and observing trajectories.

In this *model-free* setting we cannot write down $\mathcal{D}$ explicitly or feed it into a convex solver. The only way to evaluate or differentiate $L(\pi, y)$ is via sampled rollouts.

Role of RL in the saddle-point formulation. The Dual–SPMA algorithm therefore uses:

- a **dual optimizer** in y (or λ) that treats the problem as convex and uses stochastic gradients estimated from data,
- an **RL algorithm** (SPMA) as a *policy oracle*: given a fixed y, it approximately solves

$$\min_{\pi} L(\pi, y) = \min_{\pi} \Big(\langle y, d_\pi \rangle - f^\star(y) \Big)$$

  by treating $-y$ as a shaped reward and running policy optimization.

So we use RL not because the problem is non-convex in d, but because we do not have direct access to d or the model. RL is the mechanism that lets us *approximately minimize over policies* by interacting with the environment.

5. Why do we use a simple FTL / gradient update for the dual player, but SPMA for the policy player?

The two players live in very different spaces and have different structures:

Dual player (y or λ).

- Lives in a relatively low-dimensional Euclidean space (one coordinate per state–action, or one scalar constraint multiplier).
- The dual objective is *convex* (often strongly convex) and we can compute or estimate its gradient easily from occupancy estimates.
- The constraints on y are simple (e.g., box constraints, non-negativity for λ).

For such problems, classical convex optimization and online convex optimization (OCO) algorithms like Follow-the-Leader (FTL), Follow-the-Regularized-Leader (FTRL), or simple gradient ascent are natural and have strong regret/convergence guarantees.

Policy player (π).

- Lives in a huge or infinite-dimensional space: one distribution over actions per state, represented via neural network logits.
- Must satisfy probability simplex constraints at every state.
- The objective is an expected return under a shaped reward, estimated from trajectories, with high-variance gradients.

Here SPMA (Softmax Policy Mirror Ascent) is well suited because:

- It performs mirror ascent in the *logit* (dual) space with the softmax map as its mirror map.
- In the tabular case it enjoys non-asymptotic convergence guarantees that are stronger than vanilla policy gradient.
- Its loss looks similar to PPO/MDPO in the deep RL setting, making it practical to implement.

Asymmetric but natural choice. Thus:

- For the dual player we choose a simple FTL / gradient update because the space is small and convex.
- For the policy player we choose SPMA because it is designed for probability distributions / policies and has both theory and empirical support.

In words, I like to say:

> "The dual player is an ordinary convex optimization problem, so we use a simple FTL/gradient step. The policy player is a hard RL problem over probability distributions, so we use a geometry-aware method (SPMA) that is designed precisely for that setting."

6. Why do we work with occupancy measures instead of the standard 'expected return' objective in terms of π?

Let us define the *discounted occupancy measure* of a policy π:

$$d_\pi(s,a) = (1-\gamma)\,\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t \mathbf{1}\{s_t = s, a_t = a\}\right],$$

which is the normalized discounted frequency with which the pair (s,a) is visited under π.

Linearity of return in d_π. The discounted return can be written as

$$\begin{aligned} J(\pi) &= \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] \\ &= \sum_{s,a} r(s,a)\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t \mathbf{1}\{s_t = s, a_t = a\}\right] \\ &= \frac{1}{1-\gamma}\sum_{s,a} r(s,a)d_\pi(s,a) \;=\; \frac{1}{1-\gamma}\,\langle r, d_\pi\rangle. \end{aligned}$$

So *up to the constant factor* $(1-\gamma)^{-1}$, the RL objective is **linear in the occupancy measure** d_π.

Constraints like "expected discounted cost $\le \tau$" can be written similarly as $\langle c, d_\pi\rangle \le \tau$.

Convexity in occupancy space. The set $\mathcal{D}$ of all valid occupancy measures is defined by linear flow constraints (a variant of the Bellman equation) and the normalization condition $\sum_{s,a} d(s,a) = 1$. This set is a *convex polytope.* Importantly:

- each stationary policy π induces a unique $d_\pi \in \mathcal{D}$,
- conversely, each $d \in \mathcal{D}$ corresponds to at least one policy.

Now consider objectives of the form

$$f(d) = -\langle r, d\rangle + \text{convex regularizers}(d),$$

for example:

- entropy-regularized RL: $f(d) = -\langle r, d\rangle + \alpha \sum_{s,a} d(s,a) \log d(s,a)$,
- constrained safety with a convex penalty on $\langle c, d\rangle - \tau$.

Since $\mathcal{D}$ is convex and f is convex in d, the convex MDP

$$\min_{d \in \mathcal{D}} f(d)$$

is a *convex optimization problem.* This is no longer true if we parametrize directly by π; as a function of the policy parameters, $J(\pi)$ is generally *non-convex.*

Why we still talk about policies. We cannot optimize directly over d in large unknown MDPs, but d_π is the conceptual object that makes the convex structure explicit and enables the Fenchel dual formulation. In practice we:

- use RL (SPMA) to update π under shaped rewards,
- estimate d_π (or its features) from trajectories,
- use d_π in the dual update and in evaluating the convex objective.

So occupancy measures are the *bridge* between:

- the convex analysis we apply on top (Fenchel duality, saddle-point view),
- and the non-convex policy parameterization we actually implement with neural networks.

That is why the entire framework is formulated in terms of d_π, even though the algorithm itself operates on π.