Problem Setting : ST Game Model - MARL Paper

$$\left\{M := \left\{S, \{A_i\}_{i=1}^{n}, P, r, \rho, H\right\}\right\}$$

→ n agents

→ 'H' → horizon of Markov Game

P
↳ defines state transition prob function, defines game state's change over time

→ $s_h \in S$ → all system states
↳ state at horizon h

→ A_i (action space of agent i)

$A = A_1 \times A_2 \dots A_n$ (combination of entire action space of all agents)

↳ going to s_{h+1} from s_h at time step 'h+1'

& actions $a_h = \{a_{1,h}, a_{2,h} \dots a_{n,h}\}$ taken by all 'n' agents at time 'h'

next state, $s_{h+1} \sim P(\cdot / s_h, a_h)$, drawn from prob distribution defined by 'P', given the current state (s_h & actions a_h)

'r' is reward & 'r_h' is reward at time step 'h'

$r_h : S \times A \to [0, 1]$ matlab like $r_h(s_h, a_h)$ ↙ is reward that we get at a specific 'h', bounded b/w 0 & 1.

'ρ' → initial state dist

Stochastic policy 'π' $= \{\pi_h : S \to \Delta(A)\}_{h=1}^{H}$ }
↳ is a strat for agents which specifies how to choose actions probabilistically at each stage in the game

$\Delta(A)$
→ prob simplex over action space 'A' → all possible prob dists to distribute the action space A over

↙ assumes that agents choose their actions non-deterministically but by probabilities defined by this policy.

Understanding the Value function & Q-function

Value function. $V_{i,h}^{\pi}(s)$ ⇒ means value function for i^{th} player, at time step h, & state s and under the ST policy π defines the expected cumulative reward that agent 'i' receives, defined as

↓ depends on $\pi_{h:H}$

$$V_{i,h}^{\pi}(s) := E\left[\sum_{h'=h}^{H} r_{i,h'}(s_{h'}, a_{h'}) \,/\, s_h = s\right]$$

↓ expectation over ↓ cumulative reward over time from h' till total time H

Q-function for agent 'i' at time step h & state s and so on...
basically represents the expected cumulative reward that agent 'i' receives starting from state 's', defined as ⇒

$$Q_{i,h}^{\pi}(s,a) = E\left[\sum_{h'=h}^{H} r_{i,h'}(s_{h'}, a_{h'}) \,/\, s_h = a, a_h = a\right]$$

↓ dependson $\pi_{h+1:H}$

↳ This expectation is taken over future rewards, assuming that policy 'π' gets followed for all future stages

MPE defined → deterministic strategy

↓ Markov Perfect Equilibrium

$$Q_{i,h}^{\pi}(s, a^*) > Q_{i,h}^{\pi}(s, a_i, a_{-i}^*) \quad \forall a_i \neq a_i^*$$

↳ basically means that Q-value (or expected reward) of choosing equilibrium action a^* must be strictly greater than Q-value of any alt action.

→ essentially ensures that no agent deviates from their equilibrium action as all other alt actions yield lower rewards

→ Definition 2 [✓]

→ Definition 3 [✓]

→ Definition 4 [✓]

→ 2.2 [Equilibrium Selection for Normal Form Games]

$r_i : A \to \mathbb{R}$: The reward fnc for agent i (total 'n' agents), depends on the joint action space $A = A_1 \times A_2 \dots \times A_n$

→ Iterative Learning

→ these agents adjust strats iteratively based on past st outcomes

→ learning rules adjusted by Markov Chain

$$(a^{(t+1)}, \xi^{(t+1)}) \sim K_\varepsilon(\cdot, \cdot \mid a^{(t)}, \xi^{(t)})$$

- K_ε → the main transition kernel
  - ↳ defines the prob of transitioning to next state given current state.
- $a^{(t)}$ → action taken at iteration 't'
- $\xi^{(t)}$ → auxillary vars assisting in learning

→ 'ε' is the 'rate of mistakes', adds randomness in learning

Ex 1

↳ 'LLL'

→ No aux vars, [$E = \emptyset$]

→ transition probs based on agent 'i' rewards

$$K_\varepsilon(a^{(t+1)} = (a_{-i}^{+}, a_i^{t+1}) \mid a^{(t)}; \{r_i\}_{i=1}^{n}) = \frac{e^{-r_i(a_i^{(t+1)})}}{\sum_{a_i} \varepsilon^{-r_i(a_i, a_{-i}^{(t)})}}$$

→ basically actions with higher rewards are more probable / likely but not guaranteed.

→ as $\varepsilon \to 0$, we get to best response strat

→ as $\varepsilon > 0$, learning process is 'ergodic', ensuring convergence to a 'unique stationary dist'

→ Main purpose of iterative learning

introduction of 'ε' or rate of mistakes results in agents converging to a 'good' equilibrium

→ Assumption 1 (Ergodicity) [✓]

↳ predictable sys cuz as $t \to \infty$, prob of being in any state converges to a fixed value regardless of initial state.

intuitive example → system is a learning agent navigating a map of possible decisions, a lot of exploration element (agent explores & can also make mistakes). Over time, agent learns & and develops an understanding of best places to stay

guarantees stable & unbiased equilibrium selection.

Definition 5 [✓] | Definition 6 [✓]

'ε' → hidden var defined in (Ex 2)

SSE → stochastically stable equilibria → most stable NEs

→ Assumption 2 ✓

resistance 'R' is a constant such that, along with constants C_1 & C_2 ($C_1, C_2 > 0$)

$$R\left[((a,\xi)) \to (a',\xi')\right]$$

→ linked to transition probability

$$C_1\,\varepsilon^{R((a,\xi)\to(a',\xi'))} < K^{\varepsilon}\left(a',\xi' \,\middle|\, a,\xi : \{r_i\}_{i=1}^{n}\right) < C_2\,\varepsilon^{R((a,\xi)\to(a',\xi'))}$$

→ measures cost of transitioning b/w states in learning process

→ higher the value of 'R', harder the transition b/w states

→ R = 0, transition similar to when ε → 0

→ R = ∞, transition is as impossible as when $K_\varepsilon = 0$ for all ε

→ Definition 8 [✓] ✓ explained

The stochastic potential of a state that is defined by (a, ξ) is the minimum total resistance of all spanning trees that have state (a, ξ) as their 'root'

$\gamma(a,\xi) = \min_{T \in T(a,\xi)} R(T)$

'stochastic potential'

$T(a,\xi)$ → set of all spanning trees rooted at (a, ξ)

→ stochastic potential essentially indicates or that quantifies how 'easy' it is to reach state (a, ξ) from all other states in system.

Resistance of a path T is given by

$$R(T) = \sum_{(a,\xi)\to(a',\xi') \in T} R\left((a,\xi) \to (a',\xi')\right)$$

basically total resistance R(T) is the sum of resistance for all the edges in path T.

Theorem 1 → relationship b/w SSE & Stochastic Potential

↔ states with min (SP) or stochastic potential correspond to SSEs. These are the most resilient states requiring least effort to reach & hence dominate the sys.

→ Definition 9 [✓]

→ Corollary 1, 2, 3 [✓]

Pareto-optimal Outcome ; where no player can be made better off without making another player worse off. Such SSEs which result in 'POO' aren't necessarily NEs.
'POO'

* → SSEs that are both Nash equilibria & 'POO' or pareto optimal are generally preferred

Definition 10 & Algorithm 1

→ SGs generalize normal-form games by incorporating multiple stages, states & transitions b/w states.

→ goal is equilibrium selection for stochastic games

→ framework for the algo is modular, essentially builds up on K_ε

Key components:

→ K_ε governs Player's behavior, it determines how players adjust their actions & hidden var ξ at each iter based on rewards or value func

→ Actor-Critic Structure:

→ Algo essentially has 2 main steps

Step 1

Actor

Updates player action $a_h^{(t+1)}(s)$ & hidden vars $\xi_h^{(t+1)}(s)$ using K_ε (incorporating current state & rewards)

Step 2

Critic

Updates value function, $Q_{i,h}^{(t+1)}(s,a)$ for each player using a Bellman-like iteration, integrating both immediate & expected future rewards

System is modular cuz by using different K_ε, we get different equilibrium selection results.

Algo 1

→ Value functions $Q_{i,h}^{(0)}(s,a)$ initialized with immediate rewards $r_{i,h}(s,a)$

→ Terminal value $V_{i,H+1}^{(t)}(s)$ is initialized to 0 for all players 'i', all stages 'h', all states 's' & all iterations 't'.

→ Actions $a_h^{(0)}(s)$ & hidden var $\xi_h^{(0)}(s)$ are randomly initialized.

→ [Initializations]

for all t, $t \geq 0$ till total iterations;

for each stage h (in reverse order);

Step 1 - Actor

→ next action $a_h^{(t+1)}(s)$ & hidden var $\xi_h^{(t+1)}(s)$ sampled based on current values $Q_{i,h}^{(t)}(s,\cdot)$ using K_ε (This step determines players policy at this stage & state)

Step 2 - Critic

→ value functions updated, $V_{i,h}^{(t+1)}(s)$ & $Q_{i,h}^{(t+1)}(s,a)$:

$$V_{i,h}^{(t+1)}(s) = \frac{1}{t+1} \underbrace{\sum_{\tau=1}^{t+1} Q_{i,h}^{(\tau)}(s, a_h^{(\tau)}(s))}_{\text{avg of all past Q values at state 's'.}}$$

$$Q_{i,h}^{(t+1)}(s,a) = r_{i,h}(s,a) + \sum_{s'} P_h(s'|s,a) V_{i,(h+1)}^{(t+1)}(s')$$

$Q_{i,h}^{(t+1)}(s,a)$: state, action-value function

$V_{i,(h+1)}^{(t+1)}(s')$: expected return from state s