Learning Desirable Equilibria for Unknown Multi-Agent Systems

Siddharth Chandak

July 18, 2024

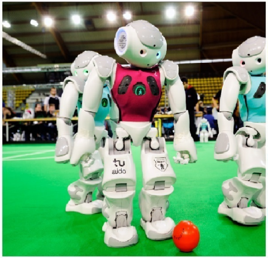
In collaboration with: Ilai Bistritz, Nick Bambos
Department of Electrical Engineering, Stanford University

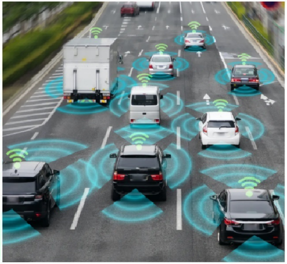
Outline

- Overview
- Game Control
- Equilibrium Bandits
- Results

Overview

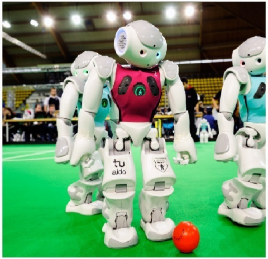
Multi-Agent Systems

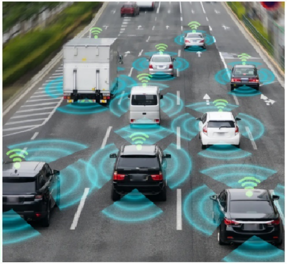




Multi-Agent Games

- Game with N agents
- Each player n takes action $\mathbf{x}_n$
- Utility (Reward): $u_n(\mathbf{x}_1, \ldots, \mathbf{x}_N)$

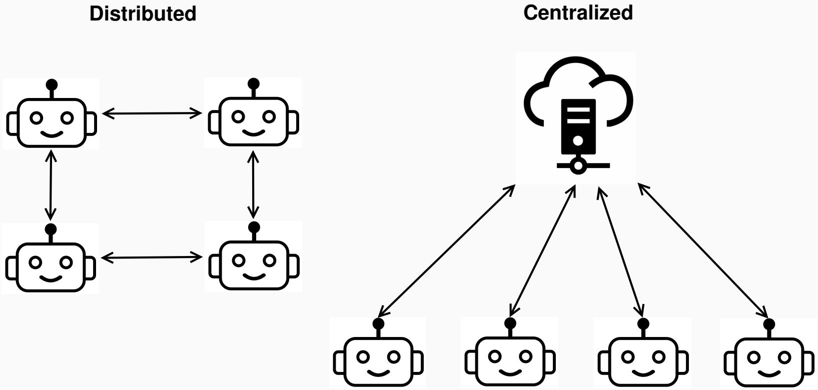




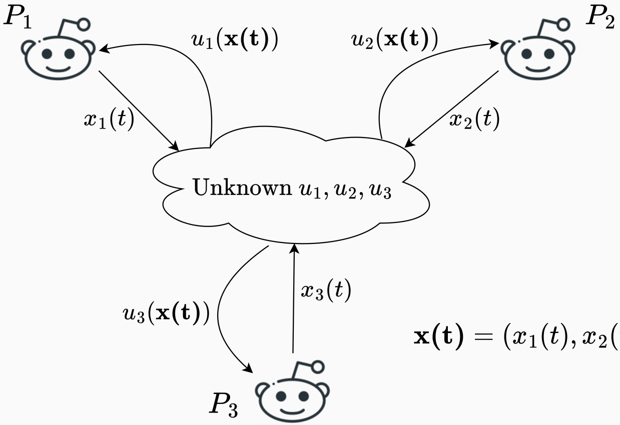
Local Objective

- **Local Objective:** Each player n wants to maximize their reward $u_n(\mathbf{x}_1, \ldots, \mathbf{x}_N)$
- Constraints:
  - Distributed System
  - Bandit Feedback
  - Limited Communication

Constraints: Distributed System

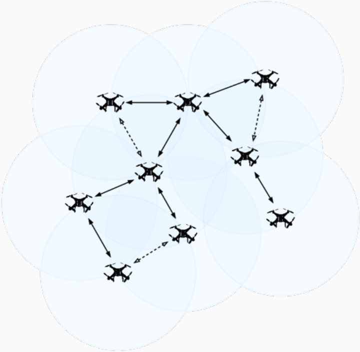


Constraints: Bandit Feedback



$\mathbf{x(t)} = (x_1(t), x_2(t), x_3(t))$

Constraints: Limited Communication



Solution Concept - Nash Equilibrium

- **Nash Equilibrium:** Action profile $\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*$ is called a Nash equilibrium if:

$$u_n(\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*, \ldots, \mathbf{x}_N^*) \geq u_n(\mathbf{x}_1^*, \ldots, \mathbf{x}_n', \ldots, \mathbf{x}_N^*),$$

  for all players n and action $\mathbf{x}_n'$.
- No benefit by unilateral deviation - no player can get a better reward if only they change their action

Example of Nash Equilibrium

		Firm 2	
		Advertise	Don't Advertise
Firm 1	Advertise	$(2, 2)$	$(6, 0)$
	Don't Advertise	$(0, 6)$	$(4, 4)$

- Each box represents profit (in \$) obtained by Firm 1 and Firm 2, respectively, under each strategy profile
- Cost of advertising $=$ \$2
- Total possible sales $=$ \$8

Example of Nash Equilibrium

Firm 2

Firm 1

	Advertise	Don't Advertise
Advertise	(2,2)	$(6, 0)$
Don't Advertise	$(0, 6)$	$(4, 4)$

- Nash Equilibrium is where both firms advertise

Convergence to NE

- Players converge to NE using gradient ascent on their rewards[1][2]
  - Completely distributed
  - Each player needs to know only their reward at each time
  - No communication between players
- If each player *slowly* changes their action to increase their reward, then the system eventually converges to a NE

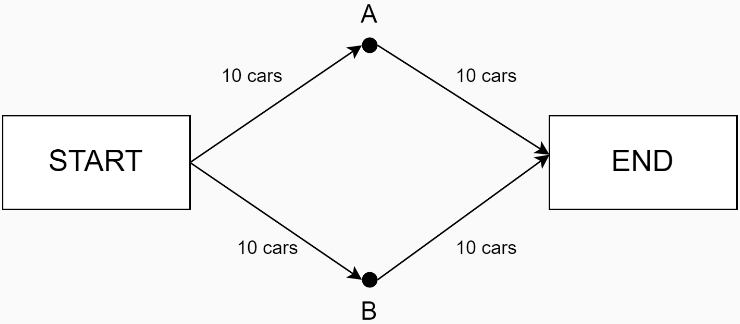
[1] recall that we are working with games with continuous actions

[2] for a class of games called monotone games

NE - good or bad?

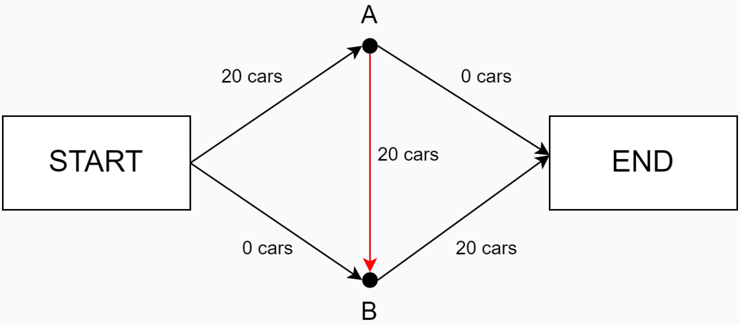
- A Nash equilibrium is not always *desirable*
- Issues:
    - Inequality
    - Inefficiency - Braess' Paradox
    - Operation Issues - Resource Allocation Games

Braess' Paradox



- 20 cars want to go from START to END
- At NE, cars are equally distributed in the two symmetric routes (top and bottom)

Braess' Paradox



- Adding an additional zero-delay road between A to B causes longer delays for every player at NE

Resource Allocation Games

- K resources
- Each player's action is $K-$dimensional, where the k^{th} dimension represents the amount of k^{th} resource they use
- Example: electricity grids and wireless channels
- At NE - often a few resources are heavily used, creating pressure on system

	Hour 1	Hour 2	$\cdots$	Hour 24
Player 1	250 W	1000 W	$\cdots$	100 W
Player 2	150 W	800 W	$\cdots$	50 W
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Player N	400 W	1500 W	$\cdots$	0 W

Game Control

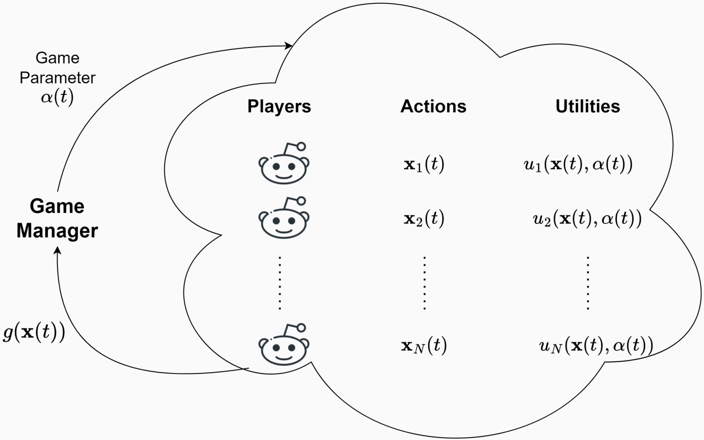
Game Parameters

- Game or multi-agent system is controlled by parameter or policy α
- Examples -
  - Toll on each road
  - Price of each resource,
  - Roads or resources available to each player
- Utility for each player $n : u_n(\mathbf{x}, \alpha)$
- NE corresponding to $\alpha : \mathbf{x}^*(\alpha) = (\mathbf{x}_1^*(\alpha), \ldots, \mathbf{x}_N^*(\alpha))$
- Consider $\alpha \in \mathcal{A}$ where $\mathcal{A}$ is a discrete and finite set

Global Objective

- Global reward - $g(\mathbf{x})$
- Problem specific
  - Sum of rewards
  - Minimum reward
  - Function of usage of each resource
- **Global Objective:** Obtain parameter α which maximizes global reward at equilibrium, i.e., find α^* such that α^* maximizes $g(\mathbf{x}^*(\alpha))$.

Global Objective



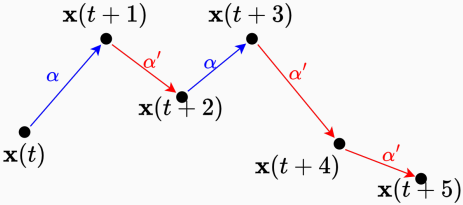
Problem Formulation

- At time t, manager sets parameter $\alpha(t)$
- Each player n observes $u_n(\mathbf{x}(t), \alpha(t))$
- Each player updates their action using gradient ascent on reward $u_n(\mathbf{x}(t), \alpha(t))$ to obtain $\mathbf{x}_n(t+1)$
- Manager observes $g(\mathbf{x}_n(t+1))$ and updates parameter

Equilibrium Bandits

Challenge

- Cannot switch at every step
  - Manager observes only $g(\mathbf{x}(t))$
  - Learns very little about reward at equilibrium $g(\mathbf{x}^*(\alpha))$

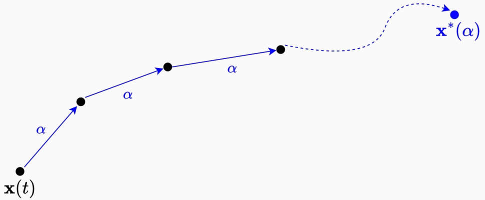


Naive Algorithm

- Manager tries each policy for a fixed number of consecutive steps t_{try}, and chooses the best policy based on the final global reward
- Gives some time to converge
- What should t_{try} be set as?
    - What if too small?
    - What if too large?

Challenge

- *Eventually* converges - how to know when?
- Want to determine if the NE for a policy will be desirable without waiting for convergence

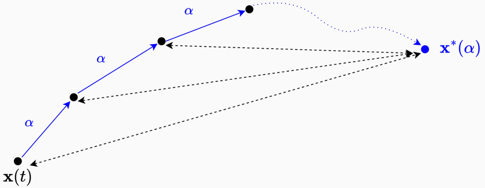


Idea: Convergence Bound

- If parameter at time t was α, then[3]

$$\|\mathbf{x}(t+1) - \mathbf{x}^*(\alpha)\| \leq \exp\left(\frac{-1}{\tau_c}\right)\|\mathbf{x}(t) - \mathbf{x}^*(\alpha)\|$$

  - τ_c : Approximate time to convergence



[3] Holds for a class of games known as strongly monotone games

Idea: Convergence Bound

- If parameter was kept as α from t to $t+\ell$ for ℓ consecutive steps,

$$\|\mathbf{x}(t+\ell) - \mathbf{x}^*(\alpha)\| \leq \exp\left(\frac{-\ell}{\tau_c}\right) \|\mathbf{x}(t) - \mathbf{x}^*(\alpha)\|$$

- Bound performance of policy at NE[4]:

$$g(\mathbf{x}(t+\ell)) - \omega e^{-\frac{\ell}{\tau_c}} \leq g(\mathbf{x}^*(\alpha)) \leq g(\mathbf{x}(t+\ell)) + \omega e^{-\frac{\ell}{\tau_c}}$$

[4] Under Lipschitz continuity assumptions on $g(\mathbf{x})$

Optimism

- Use intuition from multi-armed bandits
- **Optimism in face of uncertainty**
- Estimate of the best possible global reward for a policy (upper bound):

$$UECB = g(\mathbf{x}(t+\ell)) + \omega e^{-\frac{\ell}{\tau_c}}$$

- Try the policy with the best upper bound next

Idea: Epochs of Increasing Length

- Need to set policy for a consecutive number of times
- **Approach:** Epoch-based system: policies are changed only at ends of epochs
- Lengths of epochs increased as an policy is chosen more times
    - *Intuition:* Promising policies are given more time to converge
    - If policy α has been chosen for m epochs, then length of $(m+1)^{th}$ epoch is e^{m+1} time-steps

Upper Equilibrium Concentration Bound (UECB)

Algorithm (UECB)

For epoch $m = 1, 2, \ldots$

(1) Choose policy $\alpha_m = \arg\max_\alpha \text{UECB}_\alpha$ for $\ell_m = \exp(m_\alpha + 1)$ time-steps

(2) Update UECB:

$$\text{UECB}_{\alpha_m} = g(\mathbf{x}(t + \ell_m)) + \omega e^{-\ell_m/\tau_c}$$

End

Results

Guarantees

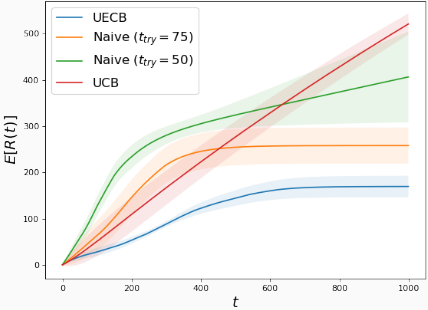
Theorem

For any instance of equilibrium bandits, UECB takes a maximum of $\widehat{T}$ time steps to identify the optimal policy α^* where

$$\widehat{T} = \mathcal{O}\left(\tau_c \sum_{\alpha \neq \alpha^*} \log\left(\frac{1}{\Delta_\alpha}\right)\right).$$

- Δ_α: Suboptimality gap - difference between performance of optimal policy and policy α.
- UECB is **orderwise optimal**

Numerical Experiments



- Naive strategy - try each action for a fixed number of steps and decide best based on that
- $R(t)$ - Regret or cumulative loss in reward

- Manager observes noisy rewards[5]:
  - Extension of above algorithm: Similar idea but more involved
  - Needs careful averaging and an additional term in bound to account for noise
- Find optimal parameter from a continuous set of parameters[6]:
  - Algorithm is based on two time-scale stochastic approximation
  - Players update their actions on a faster time-scale
  - Manager updates their policy on a slower time-scale

[5] Chandak, Bistriz, Bambos, *Equilibrium Bandits: Learning Optimal Equilibria of Unknown Dynamics*, AAMAS 2023

[6] Chandak, Bistritz, Bambos, *Learning to Control Unknown Strongly Monotone Games*, submitted to IEEE TAC

Thank You!