Tug of Peace: Distributed Learning for Quality of Service Guarantees

Siddharth Chandak, Ilai Bistritz, Nicholas Bambos

December 13, 2023

IEEE CDC 2023

Outline

- Motivation & Problem Formulation
- Tug-of-Peace Algorithm
- Results
- Proof Sketch

Motivation & Problem Formulation

Notation

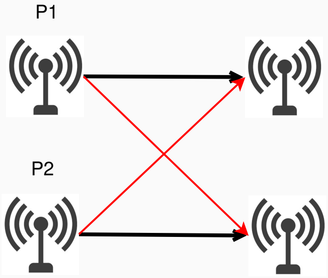
- Consider a game with N players
- Each player n takes continuous action $x_n \in \mathcal{X}_n$
- $\mathbf{x} \coloneqq (x_1, \ldots, x_N)^T$
- Receives Utility (Reward): $u_n(\mathbf{x})$

What is Quality of Service (QoS)?

- **Intuition** - Want each agent to be "sufficiently happy"
- Each agent n has their own QoS requirement λ_n
- Objective for each agent n:

$$u_n(x_1, \ldots, x_N) \geq \lambda_n$$

Example: Power Control in Wireless Networks



- Players - Transmitters
- Action - Transmission Power
- Utility - Signal-to-Interference Ratio (SIR)
- Vast literature on obtaining QoS for such games
    - Foschini et al. (1993), Yates (1995), Biguesh et al. (2011), etc.
    - Employ very specific techniques

Tug-of-War Games

- Each player takes action in $\mathbb{R}$
- **Intuition**: Increase in player 1's action reduces rewards for all other players

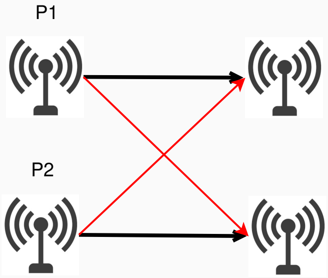
Definition 1 (Tug-of-War Game)

A game is a ToW game if the utility function is continuously differentiable and satisfies

$$\frac{\partial u_n(\mathbf{x})}{\partial x_m} < 0, \ \forall m \neq n.$$

Also $u_n(\mathbf{x}) = 0$ if $x_n = 0$ and $u_n(\mathbf{x}) \geq 0, \ \forall x$.

Application 1: Power Control in Wireless Networks

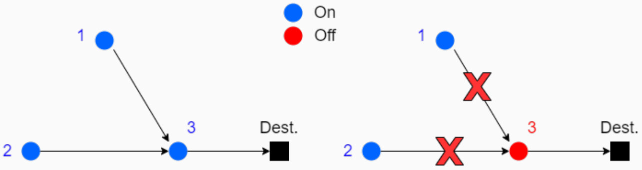


- Players - Transmitters
- Action - Transmission Power
- Utility - Signal-to-Interference Ratio (SIR)

Problem Formulation

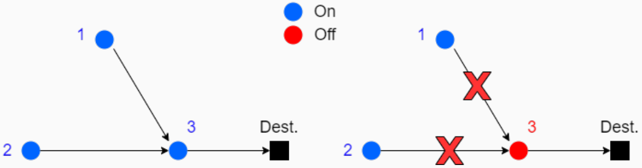
- Action set $\mathcal{X}_n$ for each player: $\mathcal{X}_n := [0, B_n] \subseteq \mathbb{R}$
- Each player chooses action $x_n(t)$ at each time $t \in \{0, 1, 2, \ldots\}$
- Observes noisy reward $y_n(t) = u_n(\mathbf{x}(t)) + M_t$ where M_t is martingale difference noise
- Wish $\mathbf{x}(\mathbf{t}) \xrightarrow{a.s.} \hat{\mathbf{x}}$ where $u_n(\hat{\mathbf{x}}) \geq \lambda_n$ for all n

Application 2: Activation in Sensor Networks



- Player: Sensors in a network
  - Collect data and also relay observations from other sensors
  - *On* (awake) or *Off* (asleep) at each time with some probability
  - When sensor is *off*: neither collects, nor relays
- Action x_n: sleeping probability for player n
- Utility for player n: $\alpha \pi_n(\mathbf{x}) - \beta B(x_n)$
  - $\pi_n(x)$: probability that player n's packets reach their destination
    - Need all sensors in route to destination to be active for packet to reach destination
  - $B(x_n)$: battery usage of player n

Application 2: Activation in Sensor Networks



- Action for player n: x_n is sleeping probability for player n
- Utility for player n: $\alpha\pi_n(\mathbf{x}) - \beta B(x_n)$
    - $\pi_n(x)$: probability that player n's packets reach their destination
    - $x_m \uparrow \implies \pi_n(\mathbf{x}) \downarrow \forall n$
    - $\boldsymbol{x_m \uparrow \implies u_n(\mathbf{x}) \downarrow \forall m \neq n}$
    - $x_m \uparrow \implies B(x_m) \downarrow$

General Setting

Why can Power Control algorithms not work for general ToW games?

- Multiple Equilibria
- Boundary Issues
- Unknown System
- Handling Noise

Goals

Problem 1

Design a distributed algorithm which requires "little" communication between agents such that $\mathbf{x}(t) \xrightarrow{a.s.} \hat{\mathbf{x}}$*, such that* $u_n(\hat{\mathbf{x}}) \geq \lambda_n$*, for all* n

Subproblem 1

$\mathbf{x}(t) \longrightarrow \mathbf{x}_*$*, where* $\mathbf{x}_*$ *is the minimal point s.t.* $u_n(\mathbf{x}_*) \geq \lambda_n$*, for all* n

Tug-of-Peace

Intuition - Iteration

Iteration:

$$x_n(t+1) = x_n(t) + \eta(t)(\lambda_n - u_n(\mathbf{x}(\mathbf{t}))).$$

Intuition:

- Increase action if receive reward lower than QoS requirement
- Decreases rewards for other players
- Other players also increase their action
- ‘Cooperative’ increase in actions leads to convergence

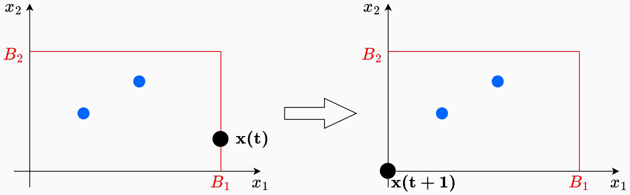
Stepsize $\eta(t)$**:**

$$\sum_t \eta(t) = \infty, \;\; \sum_t \eta(t)^2 < \infty \;\text{ and }\; \eta(t+1) < \eta(t)$$

Potential Issue - Boundaries

- $x_n(t)$ need to be inside $\mathcal{X}_n = [0, B_n]$ for all n and t
- Noise can cause iterates to go beyond boundaries - need to project iterates back into $\mathcal{X}_n$
- Denote the projection operator into $\mathcal{X}_n$ by $\Pi_{\mathcal{X}_n}$
- But this can lead to equilibrium points at boundary which do not satisfy QoS condition

Intuition - Reset at Boundary



When at boundary:

- Send *alarm* signal to every player.
- All players reset to action 0 on receipt of *alarm* signal

Intuition:

- **1-bit signal** to avoid the possibility of being stuck at boundary
- Resets iteration with a lower starting stepsize

Tug-of-Peace Algorithm (ToP)

Algorithm 1

```
Initialization: Let $x_n(0) = 0,\ \forall n$.
At timesteps $t = 0, 1, \dots,$ each player $n$
(1) Plays action $x_n(t)$ and observes a noisy reward $y_n(t)$.
(2) Updates their action as follows:
        $x_n(t+1) = x_n(t) + \eta(t)\Pi_{\mathcal{X}_n}(\lambda_n - y_n(t)).$
(3) Transmits signal $s_n = 1$ if $x_n(t+1) = B_n$, otherwise it does nothing
    (i.e., $s_n = 0$).
(4) Resets action to $0$, i.e., $x_n(t+1) = 0$ upon receiving $s_m = 1$ from
    some player $m$.
End
```

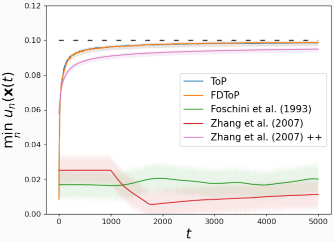
Results & Proof Sketch

Results

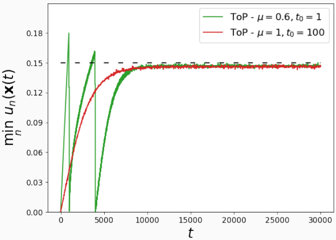
Theorem 1

1. *If the QoS requirements are feasible, then the iterates of the ToP algorithm a.s. converge to an equilibrium point* $\hat{\mathbf{x}}$ *such that* $u_n(\hat{\mathbf{x}}) \geq \lambda_n, \ \forall n$.
2. *The reset to* $\mathbf{x} = \mathbf{0}$ *happens only finitely often.*
3. *With high probability (depending on stepsize), the iterates converge to* $\mathbf{x}_*$, *where* $\mathbf{x}_*$ *is the minimal point which satifies the QoS requirements for all agents.*

Numerical Results



(a) Power Control with $N = 50$ players



(b) Sensor Activation

Proof Sketch

- **Stochastic Approximation**[1]: Iterates $\mathbf{x}(t)$ of ToP algorithm asymptotically track the solutions of the ODE

$$\dot{\mathbf{x}}(t) = \lambda - \mathbf{u}(\mathbf{x}(t))$$

- **Cooperative ODE**[2]: An ODE of form $\dot{\mathbf{x}}(t) = \mathbf{h}(\mathbf{x}(t))$, where

$$\frac{\partial h_n(\mathbf{x})}{\partial x_m} > 0$$

converges to a set of equilibria.

---

[1] Borkar (2022)

[2] Hirsch et al. (2003)

Proof Sketch

- **Domain of Attraction**[3]: $\mathbf{x} = \mathbf{0}$ lies in the domain of attraction of the minimal equilibrium point $\mathbf{x}_*$ for the ODE: $\dot{\mathbf{x}}(t) = \lambda - \mathbf{u}(\mathbf{x}(t))$
    - For any point $\hat{\mathbf{x}}$ which satisfies $u_n(\hat{\mathbf{x}}) \geq \lambda_n$ for all n, $x_{*_n} \leq \hat{x}_n$ for all n.
- **Concentration:**[4] If initiated in the domain of attraction of $\mathbf{x}_*$, the iterates $\mathbf{x}(t)$ stay in a ϵ-ball around $\mathbf{x}_*$ for all $t > T$ with high probability.

---

[3] Hirsch (1985)

[4] Thoppe et al. (2019)

Summary

Summary

- Quality of Service guarantees for Tug-of-War games
- Tug-of-Peace Algorithm
- Applications include Power Control and Sensor Activation
- Extensions for this work:
  - Asynchronous system
  - Finite-time guarantees
  - Multi-dimensional action spaces

Thank You!

Fully Distributed Tug-of-Peace

Algorithm 2

Initialization: Let $x_n(0) = 0,\ \forall n$.

At timesteps $t = 0, 1, \ldots,$ **each player** n

(1) Plays action $x_n(t)$ and observes a noisy reward $y_n(t)$.

(2) Updates their action as follows:

$$x_n(t+1) = x_n(t) + \eta(t)\Pi_{\mathcal{X}_n}(\lambda_n - y_n(t)).$$

End

Guarantees

Theorem 2

With high probability (depending on stepsize), the iterates converge to $\mathbf{x}_*$*, where* $\mathbf{x}_*$ *is the minimal point which satifies the QoS requirements for all agents.*