Tug of Peace: Distributed Learning for Quality of Service Guarantees*

Siddharth Chandak[1], Ilai Bistritz[2] and Nicholas Bambos[3]

***Abstract*— Consider N players, where the action of player n is a number in the interval $[0, B_n]$ that is interpreted as its "pull". Each player has a reward function that depends on all actions. We define Tug-of-War (ToW) games where increasing the action of one player decreases the rewards of all others. Tug-of-War games can model networking scenarios such as transmission power control and activation in sensor networks. We propose Tug-of-Peace algorithm, a simple stochastic approximation, and prove that in Tug-of-War games, it converges to a equilibrium that satisfies a target feasible Quality of Service reward vector for the players. Moreover, with high probability it converges to the "minimal pull" equilibrium. Our algorithm uses infrequent 1-bit communication between the players, but we also propose a fully distributed modification that does not require any communication at all and achieves almost the same guarantees. We then simulate our algorithms in the power control and sensor activation scenarios.**

I. INTRODUCTION

In many network scenarios, a conflict arises between the devices regarding how to share the medium. Examples are transmission power control in wireless networks, and activation probability in sensor networks. Each device has a local measure of performance, such as its throughput or energy consumption. When a device uses more power to transmit, it increases its throughput but decreases that of others. When a sensor, which collects its own data but also relays those of others, is activated with lower probability, it improves its energy consumption but worsens the packet loss rate of others. These conflicts are naturally modeled as a game between the devices.

A conflict between devices, however, does not mean that devices are selfish. The devices are programmable and run the protocol we design (e.g., in wireless protocols such as WiFi and 5G). Moreover, the devices (or players) often just have a minimum requirement of reward which we term as their Quality of Service (QoS). We wish to design a distributed algorithm that allows each player to achieve their QoS asymptotically. The challenge in designing such a protocol is that players often just observe *bandit feedback*, i.e., a noisy version of the reward at the current action profile. The players do not know their reward function, the actions of other players or the reward received by other players. Hence the challenge is to know whether to "insist" on improving their local reward or give up in favor of others.. In large-scale networks with thousands or millions of devices, this coordination challenge is the primary design bottleneck.

In this paper, we present a class of games with one-dimensional actions where increasing the action of one player decreases the rewards of others. We name this class of games *Tug-of-War (ToW)* games, due to their resemblance of players pulling a rope in opposite directions. We show that transmission power control and activation in sensor networks can both be modeled as ToW games (Section V). The importance of this class of games is twofold: it captures a common interaction in networking applications, and it allows for efficient coordination for players that are seeking to learn QoS guarantees.

We design a simple distributed algorithm (Algorithm 1) which converges with probability 1 to an action profile that satisfies the given QoS requirements for all players, under the assumption that the QoS requirements are feasible. Additionally, we show that with high probability, Algorithm 1 converges to the *minimal action profile* (Definition 2) which satisfies the QoS requirements. This algorithm requires infrequent (i.e., at finite timesteps), one-bit communications. We also present an algorithm (Algorithm 2) for the case where even one-bit communication is not possible. We show that this algorithm also converges to the minimal action profile with high probability. We name these algorithms *Tug-of-Peace* since, instead of pulling the rope as hard as they can, players learn to place the marker at a point where they can all achieve their target QoS. Finally, we demonstrate in simulations the performance of our algorithm for power control and sensor activation.

A. Related Work

QoS is a common objective in transmission power control in wireless networks [1]–[5]. This application is a special case of our model, which we study in Subsection V-A.

The algorithms presented by these works for QoS in power control exploit the form of the reward function, which they assume is known. A known form for a reward function is a reasonable assumption when designing an algorithm for a specific application. Our algorithm, on the other hand, does not need to know the reward functions, as long as the game is a ToW game.

The results of [4] provide an algorithm for power control, where assumptions are made on the knowledge available to the players. The algorithm needs each player to know the channel gain between itself and its intended receiver. [1] removes this assumption but it fails to converge in the presence of noise (as shown by [2]). The results in [2], [3] are the closest to our algorithm as they are also stochastic approximation algorithms that can handle noise and do not

*This is an extended version of the paper to be presented at IEEE Conference on Decision and Control (CDC) 2023.

[1,2,3] Department of Electrical Engineering, Stanford University, USA
[1]chandaks@stanford.edu
[2]ilaibistritz@tauex.tau.ac.il
[3]bambos@stanford.edu

assume further knowledge of the system. But they have different assumptions as they require an unbiased estimator of the inverse reward, i.e., the inverse of the Signal-to-Interference Ratio (SIR).

The analysis methods of [1]–[4] are specific to the power control wireless scenario. Their analysis depends on the eigenvalues of a matrix defined by using the channel gain matrix and the QoS vector. As an example, [2] models the problem as finding the solutions of a linear system, and its analysis then depends on the eigenvalue properties of this linear system. This model and properties are specific to that scenario and cannot be extended to our broader class of games. Other works such as [5] take a general approach, but also with certain limitations. They use the properties of monotone and sub-homogeneous maps to show the convergence of the algorithm they design. These properties are facilitated by the structure of power control games. However, designing an algorithm that satisfies these properties is not immediately obvious for other general games. Additionally, the algorithm presented here is better equipped to deal with noise than the algorithm by [5], as we show later.

A key step in the analysis in [1], [2], [5] is that their assumptions guarantee the existence of a unique equilibrium that satisfies $u_n(\boldsymbol{x}) = \lambda_n$ for all n, where $u_n(\boldsymbol{x})$ is the reward received by player n at action profile $\boldsymbol{x}$ and λ_n is their QoS requirement. Furthermore, this point is guaranteed to be globally asymptotically stable. On the other hand, these properties are not guaranteed in general ToW games. Moreover, the action sets in a ToW game are bounded which gives rise to undesirable equilibria at the boundaries. Due to the observation noise, it is tricky to guarantee to which equilibrium the algorithm would converge. As we later see in Section III, we still give guarantees on convergence, and results which show that our algorithms converge to the "best" equilibrium with high probability.

Given the general analysis and because we deal with unknown reward functions, our algorithm and its guarantees carry over to applications beyond power control with no modifications needed. One such example is activation in sensor networks, which we study in Subsection V-B. In this application, the observations are noisy and it is unreasonable to assume knowledge of the reward functions.

Beyond power control, distributed protocols with QoS guarantees were studied in [6], [7]. The work in [6] considered a multiplayer bandit scenario where, if multiple players pick the same arm, they all receive zero reward. This is a special case of a game that is different from the class of games we consider. The work in [7] considers a general discrete game, but the algorithm uses regular communication between the players. Compared to both algorithms in [6], [7], our proposed algorithm here has substantially better scalability in the number of players. Furthermore, our algorithm has two variants that either require 1-bit of infrequent communication or no communication at all.

To analyze the convergence of our algorithm, we use the Ordinary Differential Equation (ODE) approach to stochastic approximation [8] and the concept of Cooperative ODEs and Monotone Dynamical Systems [9]. Apart from their use in dynamical systems, these ODEs have been widely used in fields such as epidemiology [10], [11], social networks [12], [13] and queueing systems [14].

B. Notation

We use bold letters to denote vectors and $\mathbf{0}_N$ to denote the all-zero vector of N-dimensions. We also use $\Pi_{\mathcal{X}}$ to denote the Euclidean projection into the set $\mathcal{X}$. We further use the standard vector inequalities, where $\boldsymbol{x} \leq \boldsymbol{y}$ denotes $x_i \leq y_i$ for all i. Similarly, we use the notation $\boldsymbol{x} \in [\boldsymbol{z}, \boldsymbol{y}]$ to denote $z_i \leq x_i \leq y_i$ for all i.

II. PROBLEM FORMULATION

Consider a set of N players $\mathcal{N} = \{1, \ldots, N\}$ where each player n takes action $x_n \in \mathcal{X}_n$ where $\mathcal{X}_n \coloneqq [0, B_n] \subseteq \mathbb{R}^+$. Let $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_N$. Also define the interior of this region as $\mathcal{X}^{\text{o}} = (0, B_1) \times \ldots \times (0, B_N)$. Each player n has a reward function $u_n(\boldsymbol{x}) : \mathcal{X} \mapsto \mathbb{R}$. We now define ToW games:

Definition 1. A game is a **Tug-of-War (ToW) game** if for all players $n \in \mathcal{N}$, the reward function $u_n(\cdot)$ is continuously differentiable and satisfies the following condition:

$$\frac{\partial u_n(\boldsymbol{x})}{\partial x_m} < 0, \forall\, m \neq n \in \mathcal{N}, \boldsymbol{x} \in \mathcal{X}^{\text{o}}. \tag{1}$$

Furthermore, we assume that for all n, $u_n(\boldsymbol{x}) = 0$ if $x_n = 0$ and $u_n(\boldsymbol{x}) \geq 0, \forall n \in \mathcal{N}, \boldsymbol{x} \in \mathcal{X}$.

Intuitively, these are games where if a player increases their action keeping all else constant, then the reward of other players would drop. A broad class of games satisfying this condition are resource allocation games [15], [16]. In such games, the action taken by each player denotes the amount of resource used by that player; if a player increases their resource use then other players' reward would drop if they maintain their resource use. Definition 1 makes no assumption on the impact of a player's reward upon changing their own action (i.e., no assumption on $\partial u_n / \partial x_n$). This allows this definition to handle other types of resource allocation games, as well as games such as the application in Section V-B. We study two concrete examples of ToW games in Section V.

Each player n takes action $x_n(t)$ at each timestep $t \geq 0$. At each turn t, each player just receives a noisy variant $y_n(t)$ of its actual reward $u_n(\boldsymbol{x}(t))$. We assume that the reward $y_n(t)$ received by player n satisfies $y_n(t) = u_n(\boldsymbol{x}(t)) + M_n(t)$, where $\mathbf{M}(t) \coloneqq [M_1(t), \ldots, M_N(t)]$ is a Martingale difference sequence, i.e., $\mathbb{E}[\mathbf{M}(t)|\mathcal{F}_{t-1}] = 0$ where $\mathcal{F}_t \coloneqq \sigma(\boldsymbol{x}(s), \mathbf{M}(s), s \leq t)$. We additionally assume that $\mathbf{M}(t)$ has bounded support, i.e., $|M_n(t)| \leq \widehat{M}$, w.p. 1, for all $n \in \mathcal{N}, t \geq 0$ for some positive $\widehat{M}$. Players receive no information about the actions played by other players.

Our aim is to design a distributed algorithm such that each player n asymptotically achieves a reward of at least $\lambda_n > 0$, which we collectively denote as the Quality of Service (QoS)

vector $\boldsymbol{\lambda}$. This is not possible for all $\boldsymbol{\lambda}$ and hence we make the following feasibility assumption:

Assumption 1. *(a) There exists a $\boldsymbol{x} \in \mathcal{X}^o$ and $\delta > 0$ such that $u_n(\boldsymbol{x}) \geq \lambda_n + \delta$ for all players n.*

(b) Let $u(\cdot)$ denote the vector valued function $[u_1(\cdot), \ldots, u_N(\cdot)]$ and $Du(\boldsymbol{x})$ denote the Jacobian at $\boldsymbol{x}$. For $\overline{\boldsymbol{\lambda}}$ chosen uniformly at random in $[\boldsymbol{\lambda}, \boldsymbol{\lambda} + \boldsymbol{\delta}_N]$, the Jacobian $Du(\hat{\boldsymbol{x}})$, at points $\hat{\boldsymbol{x}}$ satisfying $u(\hat{\boldsymbol{x}}) = \overline{\boldsymbol{\lambda}}$, has no purely imaginary eigenvalues.

The first part of this assumption ensures that the QoS vector of rewards is feasible and can be achieved in the interior of the set $\mathcal{X}$. Here δ can be infinitesimally small. There could be multiple such points. Since the action of the player is a measure of "pull" (e.g., power), we prefer the equilibrium point where all elements are "minimal" (i.e., no other equilibrium exists where even one element is smaller).

The parameter δ is also needed for technical reasons. For an arbitrary QoS vector $\boldsymbol{\lambda}$, the QoS might be achieved at an equilibrium point that is not stable for our algorithm [17]. Under Assumption 1, the equilibrium point is stable with probability 1 if we take a QoS vector at random in this region. The second part of the assumption is very mild for matrices with non-zero eigenvalues, since perturbing a matrix that has purely imaginary eigenvalues adds a non zero real part to these eigenvalues [14], [17, Section 8.4]. Hence, for randomly chosen $\overline{\boldsymbol{\lambda}}$, the Jacobian $Dh(\boldsymbol{x}_*)$ does not have purely imaginary eigenvalues.

III. Distributed Algorithms

We present two algorithms in this section. The Tug-of-Peace (ToP) algorithm converges with probability 1 to a point where each player n receives at least reward λ_n, but it requires 1-bit communication between players at some timesteps. We later show that this communication is required only a finite number of times. The Fully Distributed Tug-of-Peace (FDToP) algorithm is for the case where strictly no communication is possible. The guarantees suffer in this case, but we give a result on convergence to an action vector that satisfies the QoS requirement with high probability.

A. Tug-of-Peace Algorithm

We now give an intuitive explanation for the workings of Algorithm 1.

Each player first samples a $\overline{\lambda}_n$ uniformly at random from the narrow interval $[\lambda_n, \lambda_n + \delta]$. This randomization is just a technical step and is done to ensure the almost sure stability of equilibrium points. Note that this does not affect the QoS requirement, as any point which satisfies $u_n(\boldsymbol{x}) = \overline{\lambda}_n \geq \lambda_n$ satisfies the QoS requirement for that player.

Now let us discuss the iteration performed by each player. For simplicity, first consider a case where there is no noise, so each player observes their exact reward, i.e., $y_n(t) = u_n(\boldsymbol{x}(t))$. Then the idea behind the algorithm is that each player n starts with action 0 and as they receive reward lower than its QoS requirement, they increase their action. This increase is at a rate proportional to its 'dissatisfaction', i.e., proportional to how far their reward is from the QoS requirement. By definition of ToW games, this causes a drop in reward received by other players which encourages them to increase their own actions. This 'cooperative' increase in actions eventually leads to convergence to an equilibrium point.

Even if there exists a feasible point in the interior, the noise can cause a player to reach the boundary and get stuck there. To avoid this, when a player tries to exceed the boundary, it sends a signal to other players stating that it might be stuck at the boundary. On receiving such a signal, all players return back to the action 0. This *resets* the iteration and brings it to a point that is in the domain of attraction of an equilibrium where the QoS condition is satisfied. With a lower starting stepsize after the reset, as required by Assumption 2, the probability of reaching the boundary due to noise is lower.

The signal that indicates a player is stuck at the boundary carries 1-bit of information. Furthermore, we show that these resets happen only a finite number of times with probability 1. Hence the communication overhead of Algorithm 1 is negligible. The way to implement this communication is application dependent. For example, communication between devices is more obvious in routing in sensor networks than in power control.

We need to choose appropriate stepsizes to deal with noise and the resets, which is given by the following assumption:

Assumption 2. *The stepsize sequence $0 < \eta(t) < 1$ for $t \geq 0$ satisfies the following:*

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

There may exist **multiple equilibria** which satisfy the QoS requirement. They are not all equally desirable since some require more "pull" from the players than others (e.g., more energy or power). Our algorithm selects with high probability the "minimal equilibrium", in the following sense:

Definition 2. $\boldsymbol{x}_*$ is defined as the **minimal equilibrium** if among all equilibrium points $\hat{\boldsymbol{x}}$ which satisfy $u_n(\hat{x}_n) = \overline{\lambda}_n$, $\boldsymbol{x}_*$ is the smallest component-wise, i.e., $\boldsymbol{x}_{*_n} \leq \hat{\boldsymbol{x}}_n$ for all n.

The existence of such an equilibrium point is guaranteed by Lemma 1 in the next section. Our main result gives convergence guarantees for the Tug-of-Peace algorithm:

Theorem 1. *Under assumptions 1-2, the following statements hold:*

(a) With probability 1, the iterates of Algorithm 1 converge to an equilibrium point $\hat{\boldsymbol{x}}$ which satisfies $u_n(\boldsymbol{x}) = \overline{\lambda}_n \geq \lambda_n$ for all n. Moreover, the reset to $\boldsymbol{x}(t) = \boldsymbol{0}_N$ happens only a finite number of times with probability 1.

(b) The iterates of Algorithm 1 converge to $\boldsymbol{x}_*$ with probability $1 - \varepsilon(\{\eta\})$ where $\varepsilon(\{\eta\})$ goes to zero as $\eta(0)$ goes to zero.

Here the notation $\varepsilon(\{\eta\})$ denotes the dependence of the probability on the stepsize sequence $\{\eta(t)\}$. The dependence on general $\eta(0)$ has been omitted here for simplicity. But as

an example, consider the stepsize sequence $\eta(t) = 1/(t + t_0)^\mu$ for sufficiently large $t_0 > 0$ and $0.5 < \mu \leq 1$, then $\varepsilon(\{\eta\}) = \mathcal{O}\left(t_0^{1-\mu/2} \exp\left(-C t_0^{\mu/2}\right)\right)$ for some constant $C > 0$.

Algorithm 1 Tug-of-Peace Algorithm

Initialization: Let $x_n(0) = 0,\ \forall n \in \mathcal{N}$ and $\eta(t)$ be the stepsize sequence. Let $\overline{\lambda}_n \sim \text{Unif}[\lambda_n, \lambda_n + \delta]$ for some $\delta > 0$.
At timesteps $t = 0, 1, \ldots$**, each player** $n \in \mathcal{N}$

(1) Plays action $x_n(t)$ and observes a noisy reward $y_n(t)$.
(2) Updates their action as follows:

$$x_n(t+1) = \Pi_{\mathcal{X}_n}(x_n(t) + \eta(t)(\overline{\lambda}_n - y_n(t))), \quad (2)$$

where $\Pi_{\mathcal{X}_n}$ denotes the Euclidean projection into $[0, B_n]$.
(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

B. Fully Distributed Tug-of-Peace Algorithm

If even the 1-bit communication is not possible between the players, then a player cannot signal that it might be stuck at the boundary. If just that player resets its action to zero, then the resulting action vector might still be outside the domain of attraction of a desirable equilibrium. Hence, a reset mechanism is no longer an option with no communication. For scenarios like power control, where digital communication is not yet established between the devices, 1-bit signaling requires a special design which can become an implementation burden.

Instead, in the fully distributed version of ToP we propose that a player that is stuck at the boundary just projects their action back to the boundary hoping that other players might help it reach the QoS later. This modification of the algorithm is detailed in Algorithm 2.

Algorithm 2 Fully Distributed Tug-of-Peace Algorithm

Initialization: Let $x_n(0) = 0,\ \forall n \in \mathcal{N}$ and $\eta(t)$ be the stepsize sequence. Let $\overline{\lambda}_n \sim \text{Unif}[\lambda_n, \lambda_n + \delta]$ for some $\delta > 0$.
At timesteps $t = 0, 1, \ldots$**, each player** $n \in \mathcal{N}$

(1) Plays action $x_n(t)$ and observes a noisy reward $y_n(t)$.
(2) Updates their action as follows:

$$x_n(t+1) = \Pi_{\mathcal{X}_n}(x_n(t) + \eta(t)(\overline{\lambda}_n - y_n(t))), \quad (3)$$

where $\Pi_{\mathcal{X}_n}$ denotes the Euclidean projection into $[0, B_n]$.

End

The following result gives guarantees for the Fully Distributed Tug-of-Peace algorithm.

Theorem 2. *Under assumptions 1-2, the iterates of Algorithm 2 converge with probability 1 to a point. The iterates of Algorithm 2 converge to $\boldsymbol{x}_*$, as defined in Definition 2, with probability $1 - \varepsilon(\{\eta\})$ where $\varepsilon(\{\eta\})$ goes to 0 as $\eta(0)$ goes to 0.*

The algorithm converges with probability 1. As opposed to Algorithm 1, the fully distributed version can technically converge to a *bad equilibrium*, where one or more players are stuck at the boundary, i.e., $\exists n, s.t., \hat{x}_n = B_n$. However, the second part of the theorem states that with high probability (depending on the stepsize), not only the algorithm does not converge to such a bad equilibrium, but it even converges to the best one possible. In particular, this "best" equilibrium satisfies the QoS condition, i.e., $u_n(\hat{\boldsymbol{x}}) = \overline{\lambda}_n$ for all players n **and** is "minimal".

IV. Convergence Analysis

Our convergence analysis relies on stochastic approximation and the ODE method [8]. In particular, we consider the ODE:

$$\dot{\boldsymbol{x}}(t) = h(\boldsymbol{x}(t)), \quad (4)$$

where $h_n(\boldsymbol{x}(t)) = \overline{\lambda}_n - u_n(\boldsymbol{x}(t))$. While we formally show how this ODE relates to our iterations later, it can be intuitively observed that, ignoring resets and the projections in iterations (2) and (3), these iterations are a noisy discretization of this ODE.

By definition of a ToW game, this ODE satisfies the property $\frac{\partial h_n}{x_m} > 0$ for all $n \neq m \in \mathcal{N}$. Such an ODE is called a *cooperative ODE* [18], [19]. We have already assumed that there exists an equilibrium point for this ODE in the region $\mathcal{X}^o$ (Assumption 1(a)). Then this class of ODEs have certain desirable convergence properties, which we restate in the following lemma:

Lemma 1. *The ODE given by* (4) *satisfies the following properties:*

(a) [20, Theorem 2.1] For initial conditions in an open dense set, the solutions of (4) *converge to an equilibrium.*
(b) [9, Theorem 5.6] There exists a minimal equilibrium $\boldsymbol{x}_*$ of (4) *such that any other equilibria $\hat{\boldsymbol{x}}$ satisfies $x_{*_n} \leq \hat{x}_n$ for all $n \in \mathcal{N}$.*
(c) [19] The dynamical system described by (4) *is monotone, i.e., if there are two solutions $\boldsymbol{x}(\cdot)$ and $\boldsymbol{x}'(\cdot)$ of* (4) *with $\boldsymbol{x}(0) \geq \boldsymbol{x}'(0)$, then $\boldsymbol{x}(t) \geq \boldsymbol{x}'(t)$ for all $t \geq 0$.*

The first statement of the lemma implies that solutions of the ODE (4) will converge to an equilibrium point that satisfies the QoS requirement. The last two statements together imply that any solution of the ODE initiated in the region $[\mathbf{0}_N, \boldsymbol{x}_*]$ will stay in the region $[\mathbf{0}_N, \boldsymbol{x}_*]$ for all t. The following lemma shows that the equilibrium $\boldsymbol{x}_*$ is a stable equilibrium with probability 1:

Lemma 2. *The equilibrium point $\boldsymbol{x}_*$ is a stable equilibrium point for the ODE* (4) *with probability 1. Moreover, it is a Locally Asymptotically Stable Equilibrium (LASE) [8, B.3].*

Proof. Let $u(\cdot)$ denote the vector function $(u_1(\cdot), \ldots, u_N(\cdot))^T$. Then $Dh(\boldsymbol{x}) = -Du(\boldsymbol{x})$ for all $\boldsymbol{x}$, where $Dh(\boldsymbol{x})$ and $Du(\boldsymbol{x})$ denote the Jacobians of functions h and u, respectively, computed at $\boldsymbol{x}$. Then Sard's

Lemma [21] states that the image of points for which the Jacobian $Du(\cdot)$ is nonsingular has Lebesgue measure zero, i.e., the set of points $\mathbf{z}$ such that $u(\boldsymbol{x}) = \mathbf{z}$ and $Du(\boldsymbol{x}) = 0$ has Lebesgue measure zero. Let this set with Lebesgue measure zero be denoted by $\mathcal{Z}$. Since each $\overline{\boldsymbol{\lambda}}$ is chosen uniformly from the set $[\lambda_n, \lambda_n + \delta]$, $\overline{\boldsymbol{\lambda}} \notin \mathcal{Z}$ with probability 1. This implies that $Dh(\boldsymbol{x}_*)$ is nonsingular a.s. which implies that the equilibrium point $\boldsymbol{x}_*$ is a.s. isolated. Using Assumption 1(b), the Jacobian $Dh(\boldsymbol{x}_*)$ does not have purely imaginary eigenvalues a.s. and hence the equilibrium $\boldsymbol{x}_*$ is hyperbolic a.s. Theorem 4.1.1 from [22] states that if the minimal equilibrium $\boldsymbol{x}_*$ is hyperbolic, then it is a stable equilibrium. We know that the solutions of (4) converge, and hence $\boldsymbol{x}_*$ is also a LASE. □

The implication of this lemma is that any solution of the ODE initiated in $[0, \boldsymbol{x}_*]$ will converge to the equilibrium point $\boldsymbol{x}_*$ and hence $\boldsymbol{x} = \mathbf{0}_N$ is in the domain of attraction of the LASE $\boldsymbol{x}_*$. We use this fact later in Lemma 4, but the next lemma gives a result on the convergence of the iteration (3) in the Fully Distributed Tug-of-Peace algorithm to an ODE and therefore to equilibria. This result follows from [23, Section 5.1].

Lemma 3. *Under assumptions 1-2, the following two statements hold:*

(a) The iterates of Algorithm 2 asymptotically track the solutions of the ODE

$$\dot{\boldsymbol{x}}(t) = h(\boldsymbol{x}(t)) + b(\boldsymbol{x}(t)), \tag{5}$$

with probability 1. Here $b(\boldsymbol{x}(t))$ is zero in $[0, B_1) \times \ldots \times [0, B_N)$ and $b_n(\boldsymbol{x}(t)) = -h_n(\boldsymbol{x}(t))$ if $x_n(t) = B_n$ and $h_n(\boldsymbol{x}(t)) > 0$.

(b) With probability 1, the iterates of Algorithm 2 converge to some equilibrium point $\hat{\boldsymbol{x}}$ which may be of the following form:

- *$u_n(\hat{\boldsymbol{x}}) = \overline{\lambda}_n, \hat{x}_n < B_n$ for all players n,*
- *$\exists\ n$ s.t. $\hat{x}_n = B_n$.*

The term $b(\cdot)$ is the *projection term* and is the force required to keep $\boldsymbol{x}(t)$ inside $\mathcal{X}$ at all times. The second part of the lemma states the introduction of equilibria at the boundary which may or may not satisfy our QoS condition. The projection term $b(\cdot)$ does not appear at 0, the lower boundary, because $u_n(\boldsymbol{x}) = 0$ if $x_n = 0$ and hence $h_n(\boldsymbol{x}) > 0$ at the boundary for $x_n = 0$. This implies that the driving vector field of h points inward at 0, so $b(\boldsymbol{x}) = 0$ at the lower boundary. For Algorithm 1, neither of the boundary projections have an impact. Just like the explanation above, the projection at the lower boundary has no effect and the upper boundary projections are followed by a reset, which restart the iteration.

The following lemma lower bounds the probability that the iterates of (2) or (3) will remain in a small ball around an equilibrium $\boldsymbol{x}_*$ from some time onward, if $\boldsymbol{x}(t) = \mathbf{0}_N$ for some t.

Lemma 4. *For a system satisfying assumptions 1-2, and large enough t', T, and small enough ϵ, the iterates of Algorithm 1 and 2 satisfy the following:*

$$P(\|\boldsymbol{x}(t) - \boldsymbol{x}_*\| \leq \epsilon,\ \forall t \geq t' + T + 1 \mid \boldsymbol{x}(t') = \mathbf{0}_N) \geq 1 - c(t'),$$

where the probability sequence $c(t)$ satisfies $\sum_t c(t) < \infty$.

Proof. Since $\boldsymbol{x}_*$ lies in the interior of $\mathcal{X}^o$, the boundary term in (5) does not affect its stability properties. So, $\boldsymbol{x}_*$ is a LASE for both ODEs (4) and (5) and $\mathbf{0}_N$ is in the domain of attraction for both these ODEs. As the map $h(\cdot)$ is continuously differentiable, and we have assumed that the Martingale difference sequence is bounded, we satisfy the assumptions of [24]. The high probability result then follows from Theorem 1.1 of [24]. □

The "large enough" values for t', T and $1/\epsilon$ depend on the eigenvalues of the Jacobian at $\boldsymbol{x}_*$, i.e., eigenvalues of the matrix $Dh(\boldsymbol{x}_*)$. Since we care only about convergence, the only dependence that affects us is t' as it relates to the stepsize choice. As a corollary, the result is valid from $t' = 0$ onward if the stepsize $\eta(0)$ is small enough. Moreover, $c(0)$ goes to zero as $\eta(0)$ goes to zero.

With the tools developed above, we finally give the proof for our main result in Theorem 1.

Proof of Theorem 1 (a). Let $\mathcal{A}(t')$ denote the negation of the set in the conditional probability in Lemma 4, i.e.,

$$\mathcal{A}(t') = \{\exists\ t \geq t' + T + 1 \text{ s.t.} \|\boldsymbol{x}(t) - \boldsymbol{x}_*\| \geq \epsilon\}.$$

Then we know that $P(\mathcal{A}(t') \mid \boldsymbol{x}_{t'} = \mathbf{0}_N) \leq c(t')$. The fact that $c(t)$ is summable implies that with probability 1

$$\sum_{t'} P(\mathcal{A}(t') \mid \boldsymbol{x}'_t = \mathbf{0}_N) \mathbb{I}\{\boldsymbol{x}'_t = \mathbf{0}_N\} < \infty.$$

Finally, through an extension of Borel-Cantelli Lemma [25, Corollary 5.29], we have that with probability 1:

$$\sum_{t'} \mathbb{I}\{\mathcal{A}(t'), \boldsymbol{x}(t') = \mathbf{0}_N\} < \infty.$$

This implies that $\boldsymbol{x}(t)$ from Algorithm 1 converges to $\boldsymbol{x}_*$ on the set {Resets happen infinitely often} with probability 1.

Suppose the algorithm resets back to $\mathbf{0}_N$ infinitely often, then the above argument shows that the algorithm would converge to $\boldsymbol{x}_*$. This leads to a contradiction to our assumption that infinite resets happen. And hence the algorithm resets only finitely often. Let τ denote the last such reset. Then for $t > \tau$, the iterates of (2) always stay in the interior of $\mathcal{X}$. Then the iterates a.s. asymptotically track the solutions of the ODE (4). Hence they converge to an equilibrium point of the form $\hat{\boldsymbol{x}}$ which satisfies $\hat{x}_n = \lambda_n$ for all players n. This completes the proof for Theorem 1.

□

Proof of Theorem 1 (b). As the algorithm is initialized at $\boldsymbol{x}(0) = \mathbf{0}_N$, for small enough initial stepsize $\eta(0)$, Lemma 4 can be applied for $t' = 0$. And hence with probability $1 - c(0)$, the iterates stay in the ϵ-vicinity of $\boldsymbol{x}_*$ from some

T onwards and converge to $\boldsymbol{x}_*$ as $\boldsymbol{x}_*$ is a LASE. Again, note that $c(0)$ goes to zero as $\eta(0)$ goes to zero. □

Part (b) of Lemma 3 and the proof for part (b) of Theorem 1 together complete the proof for Theorem 2.

V. Applications

We now study two problems that can be modeled using ToW games and the QoS achievability problem.

A. Power Control in Wireless Networks

In power control, the players are N transmitter-receiver pairs, so transmitter n wishes to transmit to receiver n. Each user is allocated an orthogonal channel for their communication. For each player n, the action x_n denotes the transmission power of the signal transmitted by user n. The interference experienced by receiver n is given by $I_n(\boldsymbol{x}) = \sum_{m \neq n} g_{m,n} x_m$, where $g_{m,n} > 0$ is the channel gain between the transmitter of player m and receiver of player n. Each receiver also faces additional additive Gaussian noise with variance N_0. Then the utility of each user (player) n is the Signal-to-Interference Ratio (SIR) given by:

$$u_n(\boldsymbol{x}) = \frac{g_{n,n} x_n}{N_0 + I_n(\boldsymbol{x})}. \tag{6}$$

A common objective of power control in wireless networks is to find the minimum power of transmission (or action profile) for each player such that all users satisfy their QoS requirement. Centralized control is not possible due to a range of issues including latency, communication overhead, and the additional infrastructure required.

In a distributed setting, each player is only able to get a noisy estimate of their SIR, more about which can be found in [26]. The estimation noise can be modeled as Martingale difference noise, as shown in [2]. This game can be modeled as a ToW game since the utility function satisfies:

$$\frac{\partial u_n}{\partial x_m} = -\frac{g_{n,n} g_{m,n} x_n}{(N_0 + I_n(\boldsymbol{x}))^2} < 0$$

for all $m \neq n$ and $\boldsymbol{x} \in \mathcal{X}^o$. There are multiple different assumptions on $\boldsymbol{\lambda}$ and the matrix $G = [g_{m,n}]$ in the literature [1], [2] which all result in the existence of a point that satisfies the QoS requirement. Taking the boundary B_n to be large enough for each player n would ensure that the QoS is achieved in the bounded region $[0, B_n]$. Moreover, these assumptions also result in the uniqueness of equilibrium point $\hat{\boldsymbol{x}}$ which satisfies $u_n(\boldsymbol{x}) = \lambda_n$ for all players n. Trivially this point is also the minimal equilibrium point.

Interestingly, in wireless networks, the locations of the devices, and therefore the channel gains, are typically modeled as random [1]. In this case, there is no need for Assumption 1 since the randomness in the channel gain matrix G has a similar effect on its eigenvalues.

B. Activation in Sensor Networks

Consider N sensors which communicate over a wireless network. Each sensor in this network has the task of collecting surrounding data and transmitting these observations to a destination through the communication network, i.e., they only communicate with their neighbors on the network. Hence the sensors have dual role in this setting - observing their surroundings and also relaying the observations they receive from other sensors. Players wish to collect as much data as possible, but they also wish to reduce energy consumption to save their batteries. So the sensor is active at any time only with a given probability. When activated, sensors both make observations and relay other observations. Each sensor wishes to find a probability of activation which gives them the ideal tradeoff between collecting more data and reducing energy consumption. Problems with similar formulations are studied in [27], [28].

Let the N sensors be the N players where the probability of player n to be active is denoted by p_n. Let us define the action taken by player n as $x_n = 1 - p_n$, i.e., the probability of being off. Clearly, x_n is in $[0, 1]$. We define the reward function for each player n as

$$u_n(\boldsymbol{x}) = f(P_n(\boldsymbol{x})) - \alpha + \beta x_n.$$

Here $P_n(\boldsymbol{x})$ is the probability that player n's observation was successfully transmitted to the destination. This probability depends on the actions of all players as player n's action affects the number of observations it makes and others' actions affect how many packets are relayed to the destination. $f(\cdot)$ denotes the monotonically increasing 'value' assigned to observations, which is typically concave since the marginal value of data decreases as the amount of data received increases. Finally, βx_n is the 'reward' obtained for consuming less energy thanks to being off with probability x_n. Alternatively, this can be thought of as $\beta - \beta p_n$ which is a shifted cost due to battery usage. The term $-\alpha < 0$ is just an offset term that ensures that $u_n(\boldsymbol{x}) = 0$ if $x_n = 0$.

The exact formula of $P_n(\cdot)$ can be significantly non-trivial depending on the communication network. But it is easy to justify that this game is a ToW game. If player m increases their action, then it is off with a higher probability, which decreases the times it relays observations made by player n and hence decreases $P_n(\cdot)$.

Let L be the number of data packets an active sensor transmits between two turns of our game (i.e., decision periods). This number L can be stochastic and different between sensors, but we assume it is constant for simplicity of presentation. At the end of each timestep, sensors get feedback about the number of successfully transmitted observation packets out of their L transmitted packets, which provides them with a noisy unbiased estimator of $P_n(\boldsymbol{x}(t))$.

The action x_n cannot be thought of as 'effort' or the amount of resource used (as in resource allocation games) in this case. Increasing action x_n does not necessarily increase the reward u_n for the complete domain $[0, 1]$ of x_n. Clearly $P_n(\boldsymbol{x})$ decreases and βx_n increases with increasing x_n.

Hence we can expect a peak at some point if we keep $\boldsymbol{x}_{-n}$ fixed and change x_n. The aim of the players in this game is to collect as much data as possible but they are limited by their energy consumption. Hence the "best" equilibrium in this example is one where they collect the most data, which is given by the minimal equilibrium.

In this example, the communication of the one-bit 'out-of-bounds' signal can easily be done over the existing communication network with little overhead.

VI. Simulations

In this section, we simulate the games from the previous section. Except for single-run plots, the others are the average over a 100 random realizations, which have been plotted along with the standard deviation region.

For the power control game, we randomly generate a diagonal heavy channel gain matrix G where the diagonal elements are uniformly and independently sampled from $[0.2, 0.8]$ and the non-diagonal elements are uniformly and independently sampled from $[0, 0.2]$. We set $N_0 = 0.1$. In Fig. 1a, we compare our algorithms with other algorithms designed for the power control game with $N = 100$ players and $\lambda_n = 0.1$ for each player. We plot the reward of the player with the minimum reward, i.e., the reward of the most dissatisfied player at each time. Due to the measurement noise, we assume that the rewards observed are noisy with additive Gaussian noise $\mathcal{N}(0, 0.1)$. The upper boundary B_n can be set to be large ($B_n = 1$ in this case) so that it has no effect on the problem. We use the stepsize sequence $\eta(t) = 1/(t + 100)$ for our algorithm. The dashed line represents the QoS requirement for each player.

The algorithms by [1] (*Foschini et al.* in Fig. 1a) and [2] (*Zhang et al.* in Fig. 1a) fail to converge in our case. While [1] cannot handle noise, the algorithm by [2] assumes that they have an unbiased estimator of the inverse SIR, i.e., the inverse reward, which is not the case in our formulation. The curve *Zhang et al. (2007)++* in our plot denotes the setting in their paper [2], i.e., where Gaussian noise is added to the inverse SIR. As expected, our algorithms, converge to the minimal point which satisfies the QoS requirements.

Fig. 1b plots the utilities of each player using the ToP algorithm for a power control game with $N = 4$ and different QoS requirements $\boldsymbol{\lambda} = [0.8, 1.2, 1, 0.9]^T$. We use the stepsize sequence $\eta(t) = 1/(t+10)^{0.9}$ for this example. The different dashed lines denote the QoS requirements for each player. We can see in the plot that the action taken by each of the players eventually converges to a profile that satisfies the QoS requirements for each player.

For the sensor activation game, we simulate a network with $N = 10$ sensors for this game. Each sensor has a set of multiple routes for sending packets to the destination, and we assume that a transmission is successful if there exists a route where all sensors are active. For each simulation, we randomly generate an Erdos-Renyi graph with edge probability 0.2 for the sensors. This graph dictates all possible paths from each sensor to its destination. We choose $L = 100$ packets and $f(p) = 0.8\sqrt{p}$, $\alpha = 0.8$ and $\beta = 2$.

We also studied the effects of stepsize sequence on the resulting equilibrium in the simulations for sensor activation (Fig. 2). Note that in the simulations for the power control game, the actions never got stuck at the boundary and the equilibrium is unique.

In Fig. 2a, we plot the performance of the ToP algorithm. The first plot is for a stepsize sequence which decreases slowly and has a higher initial value, i.e., $\eta(t) = 1/(t+1)^{0.6}$. In this setting, the rewards quickly increase in the beginning but one of the players reaches the boundary, causing a reset. This happens twice before the algorithm stabilizes and converges to the minimal equilibrium. The second plot is for a quickly decreasing stepsize sequence with a lower initial value ($\eta(t) = 1/(t + 100)$). In this case, the actions slowly increase and directly reach the minimal equilibrium point.

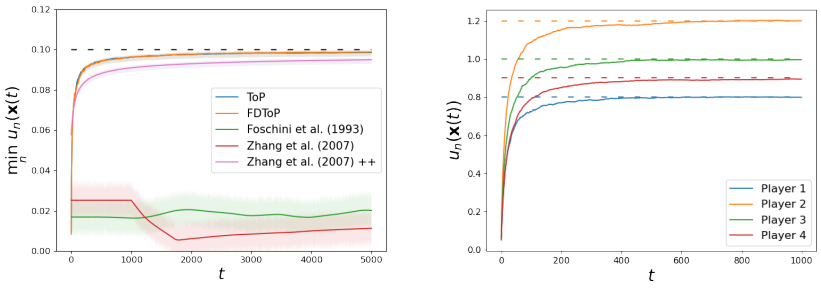
The second plot (Fig. 2b) shows the total power consumed by all sensors when they run the FDToP algorithm with different stepsizes. Stepsizes with higher initial values start with large oscillations but eventually, reach close to the minimal equilibrium point faster. On the other hand, quickly decreasing stepsizes with lower initial values are more stable but take longer to converge. Nevertheless, for all different stepsize sequences, the algorithm still converged to the same equilibrium, which was the minimal equilibrium.

In the sensor activation game, it is trivial to see that there are many more equilibrium points of our ODEs, including many at the boundary. But in the extensive simulations we performed, the algorithm always converged to the minimal equilibrium point. We tried various levels of noise (with additional additive noise), and stepsize sequences that go down very slowly (e.g., $\eta(t) = 1/\lceil t/100 \rceil^{0.51}$). Even in these conditions, we observed that both our algorithms converged to the minimal equilibrium point, despite the initial oscillations. These empirical findings suggest that in special cases of ToW games, the convergence properties can be stronger than our general theoretical guarantees.

VII. Conclusions

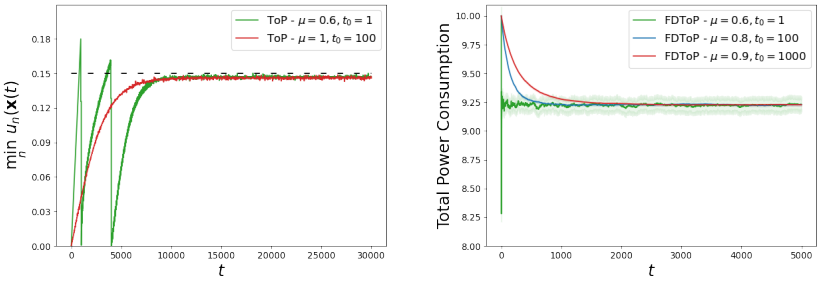
We proposed a simple stochastic approximation algorithm that players can use to converge into a point that satisfies all of their QoS requirements if such a point is feasible. We identified a class of games, called "Tug-of-War" games, and proved that for games in this class, our simple algorithm almost surelt converges to such a desirable point. Moreover, the algorithm converges to the point where the actions of the players are "minimal" with high probability, which is useful when the action represents power or energy. We model power control in wireless networks and sensor activation for data collection as ToW games and simulate our algorithm in these examples.

To make our algorithm fully distributed, we would want to generalize our analysis to asynchronous players. Another significant extension is to games where players take multidimensional actions. Such an extension would widen the scope of our algorithm, but it is unclear how the definition of ToW games can be extended to multidimensional action spaces.



(a) Comparison between algorithms (b) Performance of Tug-of-Peace Algorithm

Fig. 1: Power control game with (a) $N = 50$, (b) $N = 4$ players



(a) Effect of stepsize on ToP algorithm (b) Effect of stepsize on FDToP algorithm

Fig. 2: Sensor activation game with $N = 10$ sensors for (a) ToP, and (b) FDToP algorithm with stepsize $\eta(t) = 1/(t + t_0)^\mu$

ACKNOWLEDGMENT

The authors gratefully acknowledge funding from the Koret Foundation grant for Smart Cities and Digital Living 2030. SC is supported by the 3Com Corporation Stanford Graduate Fellowship.

REFERENCES

[1] G. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, pp. 641–646, 1993.

[2] H. Zhang, W. S. Wong, W. Ge, and P. E. Caines, "A stochastic approximation approach to the power-control problem," *IEEE Transactions on Communications*, vol. 55, no. 5, pp. 878–886, 2007.

[3] M. Biguesh and S. Gazor, "Distributed power control in cellular communication systems concerning inaccurate sinr reports," *IEEE transactions on vehicular technology*, vol. 60, no. 8, pp. 3657–3666, 2011.

[4] S. Ulukus and R. Yates, "Stochastic power control for cellular radio systems," *IEEE Transactions on Communications*, vol. 46, no. 6, pp. 784–798, 1998.

[5] R. Yates, "A framework for uplink power control in cellular radio systems," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1341–1347, 1995.

[6] I. Bistritz, T. Z. Baharav, A. Leshem, and N. Bambos, "One for all and all for one: Distributed learning of fair allocations with multi-player bandits," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 2, pp. 584–598, 2021.

[7] I. Bistritz and N. Bambos, "Queue up your regrets: Achieving the dynamic capacity region of multiplayer bandits," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022.

[8] V. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint: Second Edition*, ser. Texts and Readings in Mathematics. Hindustan Book Agency, 2022.

[9] M. W. Hirsch and H. Smith, "Monotone maps: a review," *Journal of Difference Equations and Applications*, vol. 11, no. 4-5, pp. 379–398, 2005.

[10] M. Y. Li and J. S. Muldowney, "Global stability for the seir model in epidemiology," *Mathematical Biosciences*, vol. 125, no. 2, pp. 155–164, 1995.

[11] S. Gao, L. Chen, J. J. Nieto, and A. Torres, "Analysis of a delayed epidemic model with pulse vaccination and saturation incidence," *Vaccine*, vol. 24, no. 35, pp. 6037–6045, 2006.

[12] C. Altafini, "Dynamics of opinion forming in structurally balanced social networks," *PloS one*, vol. 7, no. 6, p. e38135, 2012.

[13] ——, "Consensus problems on networks with antagonistic interactions," *IEEE transactions on automatic control*, vol. 58, no. 4, pp. 935–946, 2012.

[14] V. S. Borkar and D. Manjunath, "Charge-based control of diffserv-like queues," *Automatica*, vol. 40, no. 12, pp. 2043–2057, 2004.

[15] S. Agrawal, M. Zadimoghaddam, and V. Mirrokni, "Proportional allocation: Simple, distributed, and diverse matching with high entropy," in *International Conference on Machine Learning*. PMLR, 2018, pp. 99–108.

[16] I. Bistritz and A. Leshem, "Distributed multi-player bandits - a game of thrones approach," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[17] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential equations, dynamical systems, and an introduction to chaos*. Academic press, 2012.

[18] H. L. Smith, *Monotone dynamical systems: an introduction to the theory of competitive and cooperative systems*. American Mathematical Soc., 2008, no. 41.

[19] M. W. Hirsch, "Systems of differential equations that are competitive or cooperative ii: Convergence almost everywhere," *SIAM Journal on Mathematical Analysis*, vol. 16, no. 3, pp. 423–439, 1985.

[20] M. W. Hirsch and H. L. Smith, "Competitive and cooperative systems: A mini-review," in *Positive Systems*, L. Benvenuti, A. De Santis, and L. Farina, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 183–190.

[21] A. Sard, "The measure of the critical values of differentiable maps," *Bulletin of the American Mathematical Society*, vol. 48, no. 12, pp. 883 – 890, 1942.

[22] V. S. Borkar and S. Chandak, "Prospect-theoretic Q-learning," *Systems & Control Letters*, vol. 156, p. 105009, 2021.

[23] H. J. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. Springer Science & Business Media, 2003.

[24] G. Thoppe and V. Borkar, "A concentration bound for stochastic approximation via Alekseev's formula," *Stochastic Systems*, vol. 9, no. 1, pp. 1–26, 2019.

[25] L. Breiman, *Probability*, ser. Addison-Wesley series in statistics. Addison-Wesley Publishing Company, 1968.

[26] D. R. Pauluzzi and N. C. Beaulieu, "A comparison of snr estimation techniques for the awgn channel," *IEEE Transactions on communications*, vol. 48, no. 10, pp. 1681–1691, 2000.

[27] W. Yang, G. Chen, X. Wang, and L. Shi, "Stochastic sensor activation for distributed state estimation over a sensor network," *Automatica*, vol. 50, no. 8, pp. 2070–2076, 2014.

[28] A. Hideg, L. Blázovics, K. Csorba, and M. Gotzy, "Data collection for widely distributed mass of sensors," in *2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, 2016, pp. 000 193–000 198.