Online Learning for Load Balancing of Unknown Monotone Resource Allocation Games

Ilai Bistritz [1] **Nicholas Bambos** [1]

Abstract

Consider N players that each uses a mixture of K resources. Each of the players' reward functions includes a linear pricing term for each resource that is controlled by the game manager. We assume that the game is strongly monotone, so if each player runs gradient descent, the dynamics converge to a unique Nash equilibrium (NE). Unfortunately, this NE can be inefficient since the total load on a given resource can be very high. In principle, we can control the total loads by tuning the coefficients of the pricing terms. However, finding pricing coefficients that balance the loads requires knowing the players' reward functions and their action sets. Obtaining this game structure information is infeasible in a large-scale network and violates the users' privacy. To overcome this, we propose a simple algorithm that learns to shift the NE of the game to meet the total load constraints by adjusting the pricing coefficients in an online manner. Our algorithm only requires the total load per resource as feedback and does not need to know the reward functions or the action sets. We prove that our algorithm guarantees convergence in L2 to a NE that meets target total load constraints. Simulations show the effectiveness of our approach when applied to smart grid demand-side management or power control in wireless networks.

1. Introduction

As networks grow larger, allocating resources to users gets more complicated. This is mainly because different users have different preferences and requirements. Solving a resource allocation problem in a centralized server that knows all the parameters is infeasible for large-scale networks and also violates the users' privacy. As a result, there has been a surge of interest in distributed algorithms for resource allocation (Agrawal et al., 2018; Bistritz & Leshem, 2018; Boursier & Perchet, 2019; Mehrabian et al., 2020; Nayyar et al., 2016; Alpcan et al., 2018; Bistritz et al., 2020; Magesh & Veeravalli, 2019; Alatur et al., 2020).

In distributed resource allocation, each user makes local decisions on which resources to use. Since users share the resources, the actions of one user affect the rewards of others. For example, many users accessing the same server cause service delay for each other. Positive network effects are also possible if users all want to use popular resources. High congestion on a single resource increases the operation costs for the manager, can lead to system failures, and incurs suboptimal performance for the users. Therefore, load balancing is a key control objective for the manager that supervises the resource allocation protocol. For distributed resource allocation, designing an efficient load balancing protocol becomes far more challenging.

Game theory is a natural way to tackle distributed resource allocation (Alpcan et al., 2002; Marden & Wierman, 2013; Marden & Roughgarden, 2014). The game structure formalizes the effect of the interaction between users. The players may be cooperative, or they may be selfishly interested in maximizing their reward. In large-scale systems, obtaining performance guarantees is challenging even with cooperative players since they have no information regarding the reward functions of their peers. A Nash equilibrium (NE) aims to predict the outcome of such a repeated interaction. In monotone games (Rosen, 1965; Tatarenko & Kamgarpour, 2019; Mertikopoulos & Zhou, 2016), which are our focus here, simple distributed algorithms such as gradient descent of players on their reward functions are guaranteed to converge to a NE.

A NE does not optimize global objectives and can lead to poor performance. In resource allocation, this means that there is no guarantee on the total loads on each resource at NE. However, the manager controls the prices of the resources. By changing these prices the manager changes the game and its NE. In principle, one can design prices that guarantee a NE that balances the loads. However, this design requires knowing the reward functions of players and their action sets. This would require the manager to collect this information from the players, and then solve a large-scale optimization problem. This centralized approach violates the privacy of the users and is infeasible in large-scale networks. Even worse, the manager would need to collect

[1]Department of Electrical Engineering, Stanford University. Correspondence to: Ilai Bistritz <bistritz@stanford.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

the parameters periodically if they are time-varying, and then solve the large-scale optimization problem again.

To converge to a load-balanced NE in an unknown strongly monotone game, we propose an online learning approach. In our algorithm, the manager adjusts the pricing coefficients online and receives the total load on each resource as feedback. Such feedback is easy to monitor in practice, and also maintains user privacy. We prove that our algorithm converges in L2 to a NE that satisfies the target total load constraints. Therefore, our simple algorithm offers a theoretically principled load balancing protocol for distributed resource allocation in large-scale networks.

1.1. Notation

We use capital letters to denote random variables, lowercase letters to denote their realizations, and bold letters to denote vectors. We use the standard game-theoretic notation where $\boldsymbol{x}_{-n}$ is the vector of all actions except that of player n. We use $\mathbb{R}_+^K$ the denote the K-dimensional nonnegative orthant and use $\mathbf{1}_K$ to denote the K-dimensional ones vector. We use $\Pi_{\mathcal{A}}$ to denote the Euclidean projection into the set $\mathcal{A}$ and define $[\boldsymbol{x}]^+ = \max\{\boldsymbol{x}, \mathbf{0}\}$ (elementwise). We write $\boldsymbol{x} > 0$ if all the elements of $\boldsymbol{x}$ are positive.

2. Related Work

Distributed optimization (Nedic & Ozdaglar, 2009; Chavali et al., 2014; Molzahn et al., 2017) is concerned with optimizing a target objective distributedly over a network of agents. However, these agents are non-interacting computational units so there is no game structure between them. To cast the game as a distributed optimization with the action profile as the variable, each agent would need to know the effect of other agents on its reward function, which is infeasible. Moreover, distributed optimization entails significant communication overhead between agents.

Compared to mechanism design approaches (Parkes et al., 2004; Heydaribeni & Anastasopoulos, 2018; Deng et al., 2020), we focus on cooperative or myopic players and our manager does not elicit private information from the players but only receives the sum of actions (loads) of all players.

Our approach resembles the idea of a Stackelberg game (Maharjan et al., 2013; Fiez et al., 2020; Balcan et al., 2015; Birmpas et al., 2020), where the manager is the leader and the players are the followers (usually a single follower is considered). However, game-theoretic algorithms converge to a NE that can be inefficient from an optimization point of view. In resource allocation, this means that the equilibrium does not have load balancing guarantees. Our work provides a mechanism that provably shifts the unique NE to a point that satisfies our load balancing constraints.

Our learning framework can be viewed as a bandit learning problem where the manager is playing against a bandit that is not stochastic nor fully adversarial (Lattimore & Szepesvári, 2020). Instead, the rewards of this bandit are generated by the game dynamics. This "game bandit" also introduces a new type of noise. The goal of the manager is to steer the unique NE to a point that has load balancing guarantees. Hence, the type of feedback that the manager needs is about the behavior at NE. However, the dynamics are not at NE every turn. In fact, by changing the prices the manager perturbs the system which complicates and delays the convergence to the unique NE. Therefore, the main noise in this "game bandit" is an equilibrium noise that stems from the distance of the dynamics from the time-varying NE.

The most closely related literature to our work is that on intervention and control of games (Grammatico, 2017; Parise & Ozdaglar, 2020; 2019; Galeotti et al., 2020; Alpcan & Pavel, 2009; Mguni et al., 2019; Brown & Marden, 2017; Ferguson et al., 2021). In (Tatarenko & Garcia-Moreno, 2014) it was assumed that the manager knows the reward functions of the players, and can compute the desired NE. This is also the assumption in (Alpcan et al., 2009; Alpcan & Pavel, 2009), that considered the general approach of game-theoretic control. In (Grammatico, 2017), an aggregative game where the cost is a linear function of the aggregated action was considered. It was assumed that the manager can control the total loads the players observe, which can deviate from the true total loads. In (Marden et al., 2009; Melo, 2011; Sandholm, 2007), a similar approach to ours was considered for discrete congestion games, where the reward function of all players is the same convex function of the number of players that share the chosen resource. The congestion game considered there is a special case of monotone games that allow us to model far more general interactions and action sets. For example, it allows for models when not only the number of players that share a resource matters but also their identity (e.g., their geometric locations). Additionally, the pricing scheme in (Marden et al., 2009; Melo, 2011; Sandholm, 2007) requires the manager to know the reward functions. Control of dynamic Markov potential games was considered in (Mguni et al., 2019), where it was assumed that the manager knows the players' reward functions and can simulate the game.

One application where load balancing is crucial is demand-side management (DSM) for the electricity grid. DSM can be done by incentivizing players to delay the operation of some of their appliances (Deng et al., 2015; Nekouei et al., 2014). This is typically achieved by varying the prices of energy throughout the day such that they reflect the cost of producing energy. It has been shown in (Chen et al., 2014) that if the energy costs vary with the total energy consumed

at each hour, the peak demand decreases. However, (Chen et al., 2014) has no peak demand guarantees.

In wireless networks, load balancing takes the form of transmission power control. Game-theoretic design of single-channel power control (i.e., one-dimensional) was studied in (Alpcan et al., 2002; Zhou et al., 2020) under the wireless interference model.

3. Problem Formulation

Consider a set of players $\mathcal{N} = \{1, ..., N\}$, where player n chooses an action $\boldsymbol{x}_n = \left(x_n^1, \ldots, x_n^K\right) \in \mathcal{X}_n$ that represents how much to use from each of the K resources, where $\mathcal{X}_n \subset \mathbb{R}_+^K$ is a convex and compact set such that $\mathbf{0} \in \mathcal{X}_n$. Let $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_N$. Each player n has a utility function $u_n(\boldsymbol{x}; \boldsymbol{\alpha}) : \mathcal{X} \to \mathbb{R}$ of the form

$$u_n(\boldsymbol{x}; \boldsymbol{\alpha}) = r_n(\boldsymbol{x}) - \sum_{k=1}^{K} \alpha^k x_n^k \tag{1}$$

where $\boldsymbol{\alpha} = \left(\alpha^1, \ldots, \alpha^K\right)$ are pricing coefficients controlled by the manager of the game, as explained below. We refer to $r_n(\boldsymbol{x})$ as the reward function of player n. In many applications, $r_n(\boldsymbol{x})$ encodes the value player n has for usage vector $\boldsymbol{x}_n$, minus the cost of using these resources given the loads as determined by the actions of other players, $\boldsymbol{x}_{-n}$. However, our model allows for more general reward functions. We assume that $r_n(\boldsymbol{x})$ is twice continuously differentiable for each n. Without loss of generality, we assume that $r_n(\mathbf{0}, \boldsymbol{x}_{-n}) = 0$ for all $\boldsymbol{x}_{-n}$ and n, so players receive no reward if they do not use any resources. Our main assumption on the game structure is that it is strongly monotone, defined next:

Definition 1. Define the gradient operator of the game $F(\boldsymbol{x}) : \mathbb{R}^{NK} \to \mathbb{R}^{NK}$ as

$$F(\boldsymbol{x}) = \left(\nabla_{\boldsymbol{x}_1} r_1(\boldsymbol{x}_1, \boldsymbol{x}_{-1}), ..., \nabla_{x_N} r_N(\boldsymbol{x}_N, \boldsymbol{x}_{-N})\right). \tag{2}$$

We assume that there exists $\lambda > 0$ such that for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ we have

$$\langle \boldsymbol{y} - \boldsymbol{x}, F(\boldsymbol{y}) - F(\boldsymbol{x}) \rangle \leq -\lambda \|\boldsymbol{y} - \boldsymbol{x}\|^2 \tag{3}$$

which means that the game $\mathcal{G} = (\mathcal{N}, \{\mathcal{X}_n\}_n, \{u_n\}_n)$ is a strongly monotone game with parameter $\lambda > 0$.

Definition 1 implies that for each n and $\boldsymbol{x}_{-n}$, $r_n(\boldsymbol{x}_n, \boldsymbol{x}_{-n})$ is concave in $\boldsymbol{x}_n$. The linear term in (1) does not affect the strong monotonicity of $\mathcal{G}$ which depends only on $\{r_n\}$. Monotone games are a well-studied class of games (Facchinei & Pang, 2007) that includes strictly concave potential games as a special case (Mertikopoulos & Zhou, 2016). A strongly monotone game has a unique (pure) NE (Rosen, 1965), defined as:

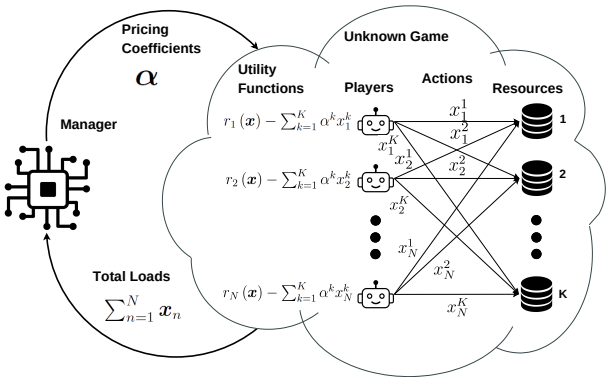


Figure 1: Control of a resource allocation game

Definition 2. A strategy profile $\left(\boldsymbol{x}_n^*, \boldsymbol{x}_{-n}^*\right) \in \mathcal{X}$ is called a pure Nash equilibrium (NE) if $u_n\left(\boldsymbol{x}_n^*, \boldsymbol{x}_{-n}^*; \boldsymbol{\alpha}\right) \geq u_n\left(\boldsymbol{x}_n, \boldsymbol{x}_{-n}^*; \boldsymbol{\alpha}\right)$ for all $\boldsymbol{x}_n \in \mathcal{X}_n$ and all $n \in \mathcal{N}$.

Our game evolves in discrete time. At turn t, given pricing coefficients $\boldsymbol{\alpha}_{t-1}$, players pick their actions $\{\boldsymbol{x}_{n,t}\}$, the manager observes the total loads $\sum_{n=1}^{N} \boldsymbol{x}_{n,t}$ as feedback and sets the new pricing coefficients $\boldsymbol{\alpha}_t$ according to (6), as discussed below. The system is illustrated in Fig. 1.

Since the game is strongly monotone, the uncontrolled players, gradually improving their rewards using gradient descent, will converge to the unique NE $\boldsymbol{x}^*(\boldsymbol{\alpha})$, which is a function of $\boldsymbol{\alpha}$. In resource allocation, a NE is typically inefficient, possibly even for all players, since the loads are not balanced. In a NE, players can keep using an overloaded resource since it is optimal given that all other players would not change their choices. An overloaded resource leads to slow service, system failures, and high operation costs, all while other resources are underutilized.

Our objective in this work is to meet the target loads $\boldsymbol{l}^*$, if feasible (as discussed below). The manager can set $\boldsymbol{l}^*$ to optimize for application-specific objectives such as system efficiency, regulations, and operation costs. In the electricity grid, the manager can be the utility company running DSM to cut the production costs of energy at peak hours (see Subsection 5.1). In wireless networks, the manager can be an access point coordinating a protocol that minimizes the power consumption of the devices and the interference in the network (see Subsection 5.2). Other examples are load balancing in data-centers (Bourke, 2001) and control of parking resources (Dowling et al., 2017).

Therefore, the manager's goal is to guarantee that the game converges to a NE that satisfies the given total loads constraints $\boldsymbol{l}^*$ by controlling the non-negative pricing coefficients $\boldsymbol{\alpha}$. Formally, the manager wants to steer the dynam-

ics $\{\boldsymbol{x}_{n,t}\}$ into the following set of load-balanced NE:

$$\mathcal{N}_{\text{opt}} = \left\{ \boldsymbol{x}^*(\boldsymbol{\alpha}^*), \boldsymbol{\alpha}^* \;\middle|\; \boldsymbol{\alpha}^* \in \mathbb{R}_+^K, \sum_{n=1}^N x_n^{*k}(\boldsymbol{\alpha}^*) = l_k^* \text{ or } \left[\sum_{n=1}^N x_n^{*k}(\boldsymbol{\alpha}^*) < l_k^* \text{ and } \alpha_k = 0 \right], \forall k \right\} \quad (4)$$

where $\boldsymbol{x}^*(\boldsymbol{\alpha}^*)$ is the NE of the game given $\boldsymbol{\alpha}^*$. We also use the notation $\mathcal{N}_{\text{opt}} = \mathcal{X}^* \times \mathcal{A}^*$ such that $\boldsymbol{\alpha}^* \in \mathcal{A}^*$ if and only if $(\boldsymbol{x}^*(\boldsymbol{\alpha}^*), \boldsymbol{\alpha}^*) \in \mathcal{N}_{\text{opt}}$.

4. Online Load Balancing Algorithm

Intuitively, if the elements of $\boldsymbol{\alpha}$ are extremely large, the total loads at NE will drop below $\boldsymbol{l}^*$, resulting in an under-utilized system. Hence we also want to guarantee that the resources are utilized, such that their load is exactly the target load if it is feasible. In (4), we only allow resource k to be under-utilized when the players do not demand more than l_k^* even when $\alpha^k = 0$. In particular, the manager will do nothing if the uncontrolled system with $\boldsymbol{\alpha}_0 = \mathbf{0}$ maintains $\sum_{n=1}^N x_n^*(\mathbf{0}) < \boldsymbol{l}^*$. The requirement in (4) can be interpreted as complementary slackness of the K constraints $\sum_{n=1}^N x_n^*(\boldsymbol{\alpha}^*) < \boldsymbol{l}^*$ with $\boldsymbol{\alpha}$ as the dual variables.

In this paper, we avoid negative $\boldsymbol{\alpha}$ since it may involve potentially unbounded payments from the manager to the players. However, if $\sum_{n=1}^N \boldsymbol{x}_n^*(\boldsymbol{\alpha}^*) = \boldsymbol{l}^*$ for some $\boldsymbol{\alpha}^* \in \mathbb{R}^K$, our algorithm can be shown to converge to a NE with total loads of exactly $\boldsymbol{l}^*$ by allowing negative $\boldsymbol{\alpha}$, with only minor modifications to our analysis.

The main challenge in guaranteeing that the dynamics converge to $\mathcal{N}_{\text{opt}}$ is that the manager does not know the game, consisting of $\{r_n\}$ and $\{\mathcal{X}_n\}$. To overcome this, we propose an online learning approach where the manager learns how to adjust $\boldsymbol{\alpha}_t$ to balance the loads by using the iteration in (6) with the control step-size sequence $\{\varepsilon_t\}$. We only assume that the manager can observe the total instantaneous loads $\sum_{n=1}^N \boldsymbol{x}_{n,t}$ at the end of turn t. Measuring the total loads is done in practice in applications such as DSM, power control, and load balancing in data-centers. This also maintains basic privacy for the players since the manager does not know their reward functions and does not even observe their individual actions.

We assume that each player runs stochastic gradient descent (SGD) to optimize its reward function, also known as gradient play. Therefore the players are myopic and do not look to manipulate the manager using dynamic strategies. This assumption is suitable for selfish players in a large-scale network or cooperative players.

In a selfish setting, players are likely to maximize their rewards using a widespread online learning algorithm such as gradient descent (Zinkevich, 2003; Lu et al., 2020; Mertikopoulos & Zhou, 2016). In particular, when the players do not know their reward functions, the small steps taken by SGD allow for gradual learning, as opposed to best-response dynamics that can have large jumps. Nevertheless, our techniques can be applied to other distributed algorithms that are guaranteed to converge to NE.

Algorithm 1 Online Load Balancing with Bandit Feedback

Initialization: Let $\boldsymbol{x}_0 \in \mathcal{X}$ and $\boldsymbol{\alpha}_0 \in \mathbb{R}_+^K$. Let $\{\eta_t\}, \{\varepsilon_t\}$ satisfy the conditions of Theorem 1.
Input: Target total load vector $\boldsymbol{l}^*$.
For each turn $t \geq 1$ **do**

1. Each player n updates its action using $\boldsymbol{g}_{n,t-1}$ and $\boldsymbol{\alpha}_{t-1}$ to approximate $\nabla_{\boldsymbol{x}_n} u_n(\boldsymbol{x}; \boldsymbol{\alpha})$:
$$\boldsymbol{x}_{n,t} = \Pi_{\mathcal{X}_n}\left(\boldsymbol{x}_{n,t-1} + \eta_{t-1}\left(\boldsymbol{g}_{n,t-1} - \boldsymbol{\alpha}_{t-1}\right)\right) \quad (5)$$
where $\Pi_{\mathcal{X}_n}$ is the Euclidean projection into $\mathcal{X}_n$.
2. The manager observes $\sum_{n=1}^N x_{n,t}^k$ for each k.
3. The manager updates the pricing coefficients using
$$\boldsymbol{\alpha}_t = \left[\boldsymbol{\alpha}_{t-1} + \varepsilon_{t-1}\left(\sum_{n=1}^N \boldsymbol{x}_{n,t} - \boldsymbol{l}^*\right)\right]^+ \quad (6)$$
where $[\boldsymbol{x}]^+ = \max\{\boldsymbol{x}, \mathbf{0}\}$ (element-wise).

End

The manager's algorithm in (6) is robust against strategic manipulations of a small group of players since each player has a negligible effect on $\sum_{n=1}^N \boldsymbol{x}_{n,t}$, which is its only way to impact (6). It is not clear if even a large group of players can manipulate (6) to increase their accumulated reward over time since any manipulated change to $\boldsymbol{\alpha}_t$ would be temporary and incurs losses to the players. Moreover, truthful reporting is not an issue in our algorithm since the players do not report anything, and the manager cannot observe their actions. Instead, the manager only needs to measure the total loads on the resources it operates.

In a cooperative setting, the behavior of the players is not something to model but to design (Bistritz & Bambos, 2020). The algorithm is used as a distributed protocol programmed into the devices in the network (e.g., WiFi). The challenge is then not to control the selfish behavior of the players, but to guide them towards the load balancing objective they do not have enough information to achieve (i.e., players do not know which $\boldsymbol{\alpha}$ would balance the total loads). For this purpose, gradient play is an efficient distributed algorithm, that has load balancing guarantees when combined with the manager's algorithm in (6).

In many scenarios, the players do not fully know their reward functions but instead learn them on the fly based on the reward values they receive (i.e., bandit feedback) or other data they collect. The gradient can then be approximated from this data using regression or other supervised machine learning techniques (Leng et al., 2020). These learning schemes make the players' actions stochastic which adds another type of noise to our analysis.

To keep our analysis general, we only require that the SGD of each player is "proper", as defined next. As a special case, our algorithm can deal with deterministic actions in the form of the standard gradient descent.

Definition 3. Define the filtration that summarizes the past

$$\mathcal{F}_t = \sigma\left(\left\{\boldsymbol{x}_{n,\tau} \,|\, \forall \tau \leq t, \forall n \in \mathcal{N}\right\}\right). \tag{7}$$

Let $g_{n,t}^k$ be the approximation of $\frac{\partial r_n(\boldsymbol{x}_n, \boldsymbol{x}_{-n})}{\partial x_n^k}$ at time t and let $\boldsymbol{g}_{n,t} = \left(g_{n,t}^1, \ldots, g_{n,t}^K\right)$ and $\boldsymbol{g}_t = \left(\boldsymbol{g}_{1,t}, \ldots, \boldsymbol{g}_{N,t}\right)$. We define a proper SGD scheme, over all players, as a scheme that outputs $\boldsymbol{g}_t$ such that for some bias sequence $\{\delta_t\}$ and $M > 0$, with probability 1:

1. $\delta_t \triangleq \left\|\mathbb{E}\left\{\boldsymbol{g}_t \mid \mathcal{F}_t\right\} - F\left(\boldsymbol{x}_t\right)\right\| \to 0$ as $t \to \infty$.
2. $\mathbb{E}\left\{\left\|\boldsymbol{g}_t\right\|^2 \mid \mathcal{F}_t\right\} \leq M$.

Note that $\boldsymbol{g}_{n,t}$ approximates $\nabla_{\boldsymbol{x}_n} r_n\left(\boldsymbol{x}\right)$ and not $\nabla_{\boldsymbol{x}_n} u_n\left(\boldsymbol{x}; \boldsymbol{\alpha}\right)$, which is approximated by $\boldsymbol{g}_{n,t} - \boldsymbol{\alpha}_t$.

Algorithm 1 details the combined scheme of the manager's iteration and the players' behavior (modeled or designed).

Intuitively, the iteration in (6) works since increasing α^k will decrease the total load on resource k. However, this simplistic intuition is missing two key aspects we discuss in the next remarks, which make it surprising that (6) works.

Remark 1 (Equilibrium Noise). We increase α_t^k to decrease the total load on resource k at NE $\sum_{n=1}^N x_n^{*k}\left(\boldsymbol{\alpha}_t\right)$. However, $\sum_{n=1}^N x_n^{*k}\left(\boldsymbol{\alpha}_t\right)$ is not available for the manager since converging to the NE $\boldsymbol{x}^*\left(\boldsymbol{\alpha}_t\right)$ typically takes (far) more than one turn. A naive algorithm would wait till the dynamics converges to NE and then run (6) on a coarse time scale that is slower than t. We prove that the manger can use $\sum_{n=1}^N \boldsymbol{x}_{n,t+1}$ instead if the step-sizes are carefully tuned, which results in a much faster and simpler algorithm.

Remark 2 (Resource Coupling). The iteration in (6) is K-dimensional, and the dimensions are coupled through the reward functions $\{r_n\}$ and the action sets $\{\mathcal{X}_n\}$. As a result, $\sum_{n=1}^N x_n^{*k}\left(\boldsymbol{\alpha}_t\right)$ does not always decrease when α^k increases. Increasing α^k will make any other resource l more attractive and may increase its total load at equilibrium, even if α^l is unchanged. Hence, the behavior of $\sum_{n=1}^N x_n^{*k}\left(\boldsymbol{\alpha}_t\right)$ as a function of $\boldsymbol{\alpha}_t$ depends on the whole $\boldsymbol{\alpha}_t$ vector and the structure of the game.

Our main result proves the convergence of Algorithm 1.

Theorem 1. *Let $\boldsymbol{l}^* > 0$. Assume that the players in our strongly monotone game (Definition 1) use a proper SGD scheme (Definition 3) with a bias sequence $\{\delta_t\}$. If the step-size sequences of the players and the manager, $\{\eta_t\}, \{\varepsilon_t\}$, are non-increasing and satisfy:*

1. *$\{\eta_t\}, \{\varepsilon_t\}$ are square summable but not summable.*
2. *$\lim_{t\to\infty} \frac{\eta_t \delta_t}{\varepsilon_t} < \infty$.*
3. *$\sum_{t=0}^{\infty} \frac{\varepsilon_t^{\frac{3}{2}}}{\eta_t^{\frac{1}{2}}} < \infty$ and $\lim_{t\to\infty} \frac{\eta_t \sum_{\tau=0}^{t-1} \varepsilon_\tau}{\sqrt{\varepsilon_t}} < \infty$.*
4. *$\lim_{t\to\infty} \frac{1 - \frac{\varepsilon_{t+1}}{\varepsilon_t}}{\eta_t} = 0$ and $\lim_{t\to\infty} \frac{\varepsilon_t}{\eta_t} < \infty$.*

Then, by running Algorithm 1,

$$\lim_{t\to\infty} \mathbb{E}\left\{\left\|\boldsymbol{x}_t - \boldsymbol{x}^*\right\|^2\right\} = 0$$

for a NE $\boldsymbol{x}^*$ where for each k, either $\sum_{n=1}^N x_n^{*k} = l_k^*$ or $\sum_{n=1}^N x_n^{*k} < l_k^*$ and $\alpha_k = 0$.

Condition 1 is standard for stochastic gradient descent. Condition 2 quantifies how fast players need to learn their reward functions (i.e., how fast δ_t has to vanish). Condition 3 states that the control update needs to be significantly slower than the action update. This way, players do not "lose track" of the rapidly changing NE. Condition 4 is merely technical and is typically implied by condition 3, as we demonstrate next.

Conditions 1-4 hold for the sequences $\eta_t = \frac{\eta_0}{(t+1)^p}$, $\varepsilon_t = \frac{\varepsilon_0}{(t+1)^q}$, $\delta_t = \frac{\delta_0}{(t+1)^d}$ such that $\frac{1}{2} < p, q \leq 1$ (condition 1), $p + d \geq q$ (condition 2) and $\frac{3}{2}q - \frac{1}{2}p > 1$ and $p + \frac{q}{2} \geq 1$ (condition 3), since by bounding the sum with an integral

$$\frac{\eta_t \sum_{\tau=0}^{t-1} \varepsilon_\tau}{\sqrt{\varepsilon_t}} \leq \eta_0 \sqrt{\varepsilon_0}\left(t+1\right)^{1-p-\frac{1}{2}q}. \tag{8}$$

Condition 4 follows since $q > p$ implies $\lim_{t\to\infty} \frac{\varepsilon_t}{\eta_t} = 0$, and $p \leq 1$ implies

$$\begin{aligned}\lim_{t\to\infty} \frac{1 - \frac{\varepsilon_{t+1}}{\varepsilon_t}}{\eta_t} &= \frac{1}{\eta_0} \lim_{t\to\infty} \left(t+1\right)^p \left(1 - \left(\frac{t+1}{t+2}\right)^q\right) \\ &\leq \frac{1}{\eta_0} \lim_{t\to\infty} \frac{\left(t+1\right)^p}{t+2} = 0. \end{aligned} \tag{9}$$

Possible values are then $q = 0.9, p = 0.6, d = 0.4$.

5. Applications

In this section, we show how our framework can be applied in two different applications.

5.1. Demand-Side Management in Electricity Grids

An immediate application of our results is demand-side management (DSM) for the electricity grid. Reducing the peak demand is the main purpose of DSM, which reduces production costs and improves the stability of the grid. DSM can also incentivize using electricity when renewable energy sources are available (e.g., during daylight).

In DSM, the resources are the K different hours of the day, and x_n^k is the amount of energy consumed by player n during hour k. The action set of player n is then

$$\mathcal{X}_n = \left\{ \boldsymbol{x}_n \,\middle|\, 0 \le x_n^k \le \gamma_n^k, \sum_{k=1}^{K} x_n^k \le \gamma_n, \forall k \right\} \tag{10}$$

where γ_n^k and γ_n are the maximal amount of energy player n may need at hour k and throughout the day, respectively. A common design for the reward function of player n is (Chen et al., 2014; Deng et al., 2015; Ma et al., 2014):

$$r_n(\boldsymbol{x}) = \sum_{k=1}^{K} \left(v_n^k\left(x_n^k\right) - P_k\left(x_n^k, \sum_{m=1}^{N} x_m^k\right)\right) \tag{11}$$

where $v_n^k\left(x_n^k\right)$ represents the value player n gives to x_n^k units of energy at hour k and $P^k\left(x_n^k, \sum_{m=1}^{N} x_m^k\right)$ is the baseline price for x_n^k units of energy at hour k. We assume that there exist $v_0 > 0$ and $x_0 \ge 0$ such that for all n, k:

$$\frac{\partial^2 v_n^k(x)}{\partial x^2} \le -v_0, \forall x \le x_0 \text{ and } \frac{\partial^2 v_n^k(x)}{\partial x^2} \le 0, \forall x > x_0 \tag{12}$$

which models the diminishing returns of the value of energy since near zero it increases faster. Every strongly concave function satisfies this assumption, as do $v_n^k(x) = \log(1+x)$ with $v_0 = \frac{1}{2}, x_0 = 1$ or $v_n^k(x) = \sqrt{x}$ with $v_0 = x_0 = \frac{1}{2}$. Let $s_k = \sum_{m=1}^{N} x_m^k$. The common design in the literature (Chen et al., 2014; Deng et al., 2015; Ma et al., 2014) uses the following pricing function for some $a_n > 0$ and $b_n \ge 0$

$$P_k\left(x_n^k, s_k\right) = a_k x_n^k s_k^2 + b_k x_n^k s_k. \tag{13}$$

It was shown in (Chen et al., 2014) that the above DSM game is strictly monotone with P_k as the cost functions of the game, so $v_n^k\left(x_n^k\right) = 0$ for all n and k. Our assumptions on $\left\{v_n^k\right\}$ make the game strongly monotone with parameter $\lambda > 0$ (see Definition 1). Our algorithm can be programmed into the energy consumption scheduler (ECS) of each consumer which chooses $\boldsymbol{x}_{n,t+1}$ a day ahead in response to the announced $\boldsymbol{\alpha}_t$ for that day. The ECS learns on the fly the preferences of its user, as encoded in $\left\{v_n^k\left(x_n^k\right)\right\}_{k=1}^{K}$, based on the feedback that the user occasionally provides.

5.2. Power Control in Wireless Networks

Another key domain our results can be applied to is that of power control in wireless networks (Alpcan et al., 2002; Candogan et al., 2010; Zhou et al., 2020; Bambos, 1998). In this application, the players are N communication links and the manager is a close-by base station or access point. The resources are K non-overlapping frequency channels and x_n^k is the transmission power of link n on channel k. The action set of link n is then

$$\mathcal{X}_n = \left\{ \boldsymbol{x}_n \,\middle|\, x_n^k \ge 0, \sum_{k=1}^{K} x_n^k \le P_n, \forall k \right\} \tag{14}$$

where P_n is total transmission power of link n. The interference link n experiences on channel k is given by $I_n^k(\boldsymbol{x}) = \sum_{m \ne n} g_{m,n}^k x_m^k$ where $g_{m,n}^k$ is the channel gain between the transmitter of link m and the receiver of link n, for channel k. The channel gains are usually modeled as random variables with expectations that are inversely proportional to the squared distance between the devices. A common reward function for each link n is its achievable throughput, treating interference as noise:

$$r_n(\boldsymbol{x}) = \sum_{k=1}^{K} \log_2\left(1 + \frac{g_{n,n}^k x_n^k}{N_0 + I_n^k(\boldsymbol{x})}\right) \tag{15}$$

where N_0 is the variance of the Gaussian noise.

It has been shown that in the low-interference regime (i.e., $\sum_{m \ne n} g_{m,n}^k$ is small compared to $g_{n,n}^k$ for all k and n), the power control game is strongly monotone (Scutari et al., 2014; Alpcan et al., 2009). In this game, if other players use higher transmission powers, player n would also be compelled to do so. Consequently, the NE of the power control game is typically not energy-efficient.

The manager can obtain $\sum_n x_n^k$ as feedback if each link transmits a pilot sequence with power $\tilde{x}_n^k = \rho_0 \frac{x_n^k}{g_{n,0}^k}$ where $g_{n,0}^k$ is the channel gain between transmitter n and the manager for channel k, and ρ_0 is a normalization factor. Then $\sum_n x_n^k$ is simply the interference the manager measures. Alternatively, the manager can define the interference at its location, $\sum_n g_{n,0}^k x_n^k$, as the total load on channel k. Mathematically, this amounts to rescaling $\mathcal{X}_n$ such that the action of player n is $\tilde{x}_n^k = g_{n,0}^k x_n^k$ which is one-to-one with x_n^k.

Our algorithm can steer the dynamics to a NE that achieves better throughput than that of the uncontrolled system while using much less power. Such a NE will save battery for the devices while maintaining low total interference to external communication systems, which is especially important for cognitive radios (Scutari et al., 2014). To implement our algorithm, the manager simply broadcasts $\boldsymbol{\alpha}_t$ to the players over the wireless channel. Broadcasting $\boldsymbol{\alpha}_t$ does not harm energy efficiency since the manager is connected to the electricity grid so it is not battery-limited.

6. Convergence Analysis

In this section, we explain our proof strategy by breaking it into lemmas. The proofs are postponed to the appendix. In a strongly monotone game, the (stochastic) gradient play converges to a unique NE for constant pricing coefficients $\boldsymbol{\alpha}$. However, in our algorithm, the pricing coefficients $\boldsymbol{\alpha}_t$ are varying in response to the actions of the players. Therefore, the game changes, and its unique NE $\boldsymbol{x}^*(\boldsymbol{\alpha}_t)$ changes as well as a function of $\boldsymbol{\alpha}_t$. As intuition suggests, changing $\boldsymbol{\alpha}_t$ by a little bit does not change $\boldsymbol{x}^*(\boldsymbol{\alpha}_t)$ by much (Lemma 3). Hence, the rate of change of $\boldsymbol{x}^*(\boldsymbol{\alpha}_t)$ is controlled by the step-size sequence $\{\varepsilon_t\}$. However, if the NE changes too fast, the players cannot trace it closely enough and would never converge to the desired NE (or to any point). The idea behind Theorem 1 is that the search for $\boldsymbol{\alpha}$ in (6) is a stochastic approximation iteration, where the main noise is the distance of $\boldsymbol{x}_{t+1}$ from the NE $\boldsymbol{x}^*(\boldsymbol{\alpha}_t)$, which we call the "equilibrium noise" (see Remark 1). In Lemma 5 we show how to tune $\{\varepsilon_t\}$ and $\{\eta_t\}$ such that this "equilibrium noise" can be averaged over time, leading to a load-balanced NE, i.e., a point in the set $\mathcal{N}_{\text{opt}}$ from (4).

6.1. Structural Properties of the Equilibria

In this subsection, we prove static properties of the desired NE, to which our algorithm will be shown to converge.

While the NE of the game is unique, $\mathcal{N}_{\text{opt}}$ does not have to be a singleton since multiple $\boldsymbol{\alpha}^*$ can lead to the same $\boldsymbol{x}^*$ vector. Additionally, the existence of the NE by itself does not guarantee that $\mathcal{N}_{\text{opt}}$ is not empty since it imposes a constraint on the NE. To prove results about the set $\mathcal{N}_{\text{opt}}$ (e.g., non-emptiness), it is useful to think of (4) in the language of variational inequalities (Facchinei & Pang, 2007).

Definition 4. Define $F(\boldsymbol{x},\boldsymbol{\alpha}):\mathbb{R}^{NK}\times\mathbb{R}^K_+\to\mathbb{R}^{NK}$ as

$$F(\boldsymbol{x},\boldsymbol{\alpha})=\left(\nabla_{\boldsymbol{x}_1}r_1(\boldsymbol{x})-\boldsymbol{\alpha},\ldots,\nabla_{\boldsymbol{x}_N}r_N(\boldsymbol{x})-\boldsymbol{\alpha}\right). \tag{16}$$

Define the variational inequality (VI) such that for all $\boldsymbol{x}\in\mathcal{X}$ and all $\boldsymbol{\alpha}\in\mathbb{R}^K_+$

$$\langle \boldsymbol{x}-\boldsymbol{x}^*, F(\boldsymbol{x}^*,\boldsymbol{\alpha}^*)\rangle+\left\langle \boldsymbol{\alpha}-\boldsymbol{\alpha}^*, \sum_{n=1}^N \boldsymbol{x}_n^*-\boldsymbol{l}^*\right\rangle\le 0 \tag{17}$$

and let $\mathcal{S}$ be the set of its solutions, such that $(\boldsymbol{x}^*,\boldsymbol{\alpha}^*)\in\mathcal{S}$ if and only if (17) holds for all $\boldsymbol{x}\in\mathcal{X}$ and all $\boldsymbol{\alpha}\in\mathbb{R}^K_+$.

The next lemma shows that $\mathcal{S}$ is precisely the set $\mathcal{N}_{\text{opt}}$ of load-balanced NE, hence motivating the VI in (17).

Lemma 1. *$\mathcal{N}_{opt}=\mathcal{S}$.*

The next lemma shows that there exist such load-balanced NE and that the set of solutions is compact.

Lemma 2. *$\mathcal{N}_{opt}$ is non-empty and compact.*

The next lemma shows that the NE changes by a small amount if $\boldsymbol{\alpha}$ changes by a small amount. This is important for our main result since the manager changes $\boldsymbol{\alpha}_t$ every iteration, so the changes in $\boldsymbol{x}^*(\boldsymbol{\alpha}_t)$ need to be well-behaved to allow for the players to stay on the path to a load-balanced equilibrium.

Lemma 3. *Let $\boldsymbol{x}^*(\boldsymbol{\alpha})$ be the unique NE given $\boldsymbol{\alpha}\in\mathbb{R}^K_+$. Then, there exists a constant $L>0$ such that for all $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2\in\mathbb{R}^K_+$*

$$\|\boldsymbol{x}^*(\boldsymbol{\alpha}_2)-\boldsymbol{x}^*(\boldsymbol{\alpha}_1)\|\le L\|\boldsymbol{\alpha}_2-\boldsymbol{\alpha}_1\| \tag{18}$$

i.e., $\boldsymbol{x}^*(\boldsymbol{\alpha})$ is L-Lipschitz continuous.

The next lemma shows that the total loads vector at equilibrium is monotone in $\boldsymbol{\alpha}$. This monotone behavior could have been sufficient to show that (6) converges if the game would have always been at equilibrium, i.e., $\boldsymbol{x}_{t+1}=\boldsymbol{x}^*(\boldsymbol{\alpha}_t)$ for all t so there is no equilibrium noise. Furthermore, this lemma shows that while $\mathcal{N}_{\text{opt}}$ may include more than one point, all of them share the same strategy profile $\boldsymbol{x}^*$.

Lemma 4. *Let $\boldsymbol{x}^*(\boldsymbol{\alpha})$ be the unique NE given $\boldsymbol{\alpha}\in\mathbb{R}^K_+$. Let $\lambda>0$ be the parameter from Definition 1. Then, for any $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2\in\mathbb{R}^K_+$,*

$$\sum_{k=1}^K\left(\alpha_2^k-\alpha_1^k\right)\sum_{n=1}^N\left(x_n^{*k}(\boldsymbol{\alpha}_2)-x_n^{*k}(\boldsymbol{\alpha}_1)\right)\le -\lambda\|\boldsymbol{x}^*(\boldsymbol{\alpha}_2)-\boldsymbol{x}^*(\boldsymbol{\alpha}_1)\|^2. \tag{19}$$

Furthermore, $\boldsymbol{x}^*(\boldsymbol{\alpha}_1)=\boldsymbol{x}^*(\boldsymbol{\alpha}_2)$ for every $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2\in\mathcal{A}^*$ (i.e., $(\boldsymbol{x}^*(\boldsymbol{\alpha}_1),\boldsymbol{\alpha}_1),(\boldsymbol{x}^*(\boldsymbol{\alpha}_2),\boldsymbol{\alpha}_2)\in\mathcal{N}_{opt}$).

6.2. Stochastic Approximation with Equilibrium Noise

In this subsection, we analyze the convergence of our algorithm to the equilibria discussed in Subsection 6.1. While the VI approach is useful to prove structural results for the equilibria, it does not provide any algorithmic insight and cannot be used to analyze the stochastic dynamics. Instead, our algorithm is based on stochastic approximation. The main source of noise in our stochastic approximation is the equilibrium noise (see Remark 1) that emerges since the manager changes $\boldsymbol{x}^*(\boldsymbol{\alpha}_t)$ before the players can converge to this NE. This type of noise is unique to our scenario and requires proving convergence from scratch using martingale techniques (Robbins & Siegmund, 1971).

The following lemma bounds the mean squared error of the equilibrium noise and is the main tool to prove Theorem 1.

Lemma 5. *Assume that the players use a proper SGD scheme (Definition 3) with a bias sequence $\{\delta_t\}$. Let $\{\eta_t\},\{\varepsilon_t\}$ be such that conditions 1-4 of Theorem 1 hold. Then, there exists $A>0$ such that for all t*

$$\mathbb{E}\left\{\|\boldsymbol{x}_{t+1}-\boldsymbol{x}^*(\boldsymbol{\alpha}_t)\|^2\right\}\le A\frac{\varepsilon_t}{\eta_t}. \tag{20}$$

7. Simulation Results

In this section, we provide numerical simulations that include the two applications discussed in Section 5. In the first two experiments, to simulate a vanishing estimation error (Definition 3), at iteration t, Gaussian noise with mean $\delta_t = \frac{0.25}{(t+1)^{0.4}}$ and variance $\sigma^2 = \frac{1}{4}$ was added to the gradients of each player. To quantify the effectiveness of our algorithm, we compare it to the uncontrolled system where $\alpha^k = 0$ for all k. For all experiments, we used the step-size sequence $\eta_t = \frac{\eta_0}{(t+1)^{0.6}}$ and the control step-size sequence $\varepsilon_t = \frac{\varepsilon_0}{(t+1)^{0.9}}$ with different values of η_0, ε_0. We ran 100 realizations for each experiment and plotted the average result along with the standard deviation region, which was always small. In each realization, $\left\{\alpha_0^k\right\}$ and $\left\{x_{n,0}^k\right\}$ were chosen uniformly and independently at random on $[0, 2]$ and $[0, 0.1]$, respectively. In all experiments, $\sum_n x_{n,t}^k$ was rarely above the uncontrolled load, and our algorithm even improved the overshoot significantly.

In Fig. 2a, we simulated the power control game from Subsection 5.2 with $N = 300$ links and $K = 3$ channels. We used $\eta_0 = 0.03$ and $\varepsilon_0 = 0.8$. The location $\boldsymbol{y}_n^1$ of transmitter n was chosen uniformly at random on a 2D square of area $2N$. The location $\boldsymbol{y}_n^2$ of receiver n was chosen uniformly at random on a 2D disk with radius one around $\boldsymbol{y}_n^1$. The channel gains were then given by $g_{n,m}^k = \frac{a_{n,m,k}}{\|\boldsymbol{y}_n^1 - \boldsymbol{y}_m^2\|^2}$ where $\{a_{n,m,k}\}$ were chosen uniformly and independently at random on $[0.5, 1.5]$. The target total transmission powers were $\boldsymbol{l}^* = \{60, 75, 90\}$, representing constraints on the interference caused to different external systems operating in these frequencies. We used $N_0 = 0.001$ in (15) and the total transmission power was $P_n = 1$ for each n.

Both $\boldsymbol{\alpha}_t$ and $\sum_{n=1}^N \boldsymbol{x}_{n,t}$ converged in less than 750 iterations in all realizations. The NE of the uncontrolled system uses total transmission powers that are higher by 9.4%, 31.3%, and 64.0% for $k = 1, 2, 3$, respectively. The total throughput $\sum_{n=1}^N r_n(\boldsymbol{x}_t)$ at $t = 750$ using our algorithm was 1184.9 ± 74.6, while that of the uncontrolled system was 1027.8 ± 71.5. We conclude that using our algorithm, one can increase the total throughput in the network while saving energy by reducing the average transmission power and by extension the interference.

In Fig. 2b, we simulated the DSM game from Subsection 5.1, with $a^k = 0.001$ and $b^k = 0$ for all k. The number of players was $N = 1000$ and the resources were the $K = 24$ hours of the day. We used $\eta_0 = 0.05$, $\varepsilon_0 = 0.2$. The value functions were given by $v_n^k\left(x_n^k\right) = \omega_n^k x_n^k - \frac{\beta}{2}\left(x_n^k\right)^2$, the maximum required energy per hour for player n at hour k was $\gamma_n^k = \frac{\omega_n^k}{\beta}$ for $\beta = 0.3$ and the maximal total energy required over the day was $\gamma_n = 2.4$ for all n. Hence, ω_n^k encodes the value of energy of user n at hour k. The expected values of ω_n^k were chosen to match the demand for electricity throughout the day:

$$\boldsymbol{\omega} = \begin{bmatrix} 1.1, 1, 1, 1, 1, 1.2, 1.3, 1.5, 1.5, 1.5, 1.6, 1.6 \\ 1.5, 1.5, 1.5, 1.5, 1.6, 1.8, 1.7, 1.6, 1.6, 1.5, 1.3, 1 \end{bmatrix}$$

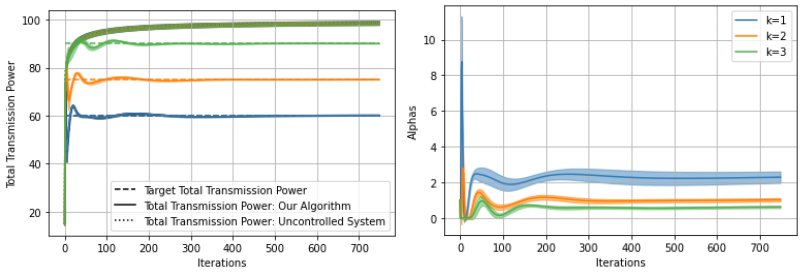
and the values of ω_n^k were generated i.i.d. between players and hours using a Gaussian distribution with mean ω^k and variance 0.5 for all k. The target loads $\boldsymbol{l}^*$ were chosen to be a factor of their corresponding total load in the uncontrolled system, with the factors given by

$$\begin{bmatrix} 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 0.85, 0.7, 0.7, 0.7, 0.7 \\ 0.7, 0.7, 0.7, 0.7, 0.5, 0.5, 0.5, 0.7, 0.85, 1, 1, 1 \end{bmatrix}.$$

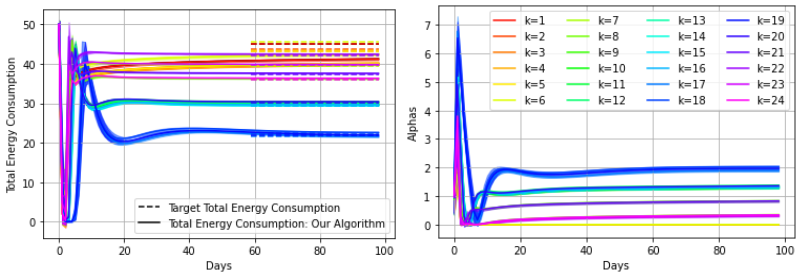
In the uncontrolled system, players are subjected to the pricing scheme of (Deng et al., 2014), given in (13).

Both $\boldsymbol{\alpha}_t$ and $\sum_{n=1}^N \boldsymbol{x}_{n,t}$ converged in less than 100 iterations in all 100 realizations. The total loads for $k > 6$ converged to l_k^* with $\alpha^k > 0$ and $k = 1, \ldots, 6$ converged to slightly below l_k^* with $\alpha^k = 0$, in all 100 realizations. The social welfare $\sum_{n=1}^N r_n(\boldsymbol{x}_t)$ at $t = 100$ using our algorithm was 744.7 ± 5.3, compared to 208.9 ± 4.1 for the uncontrolled system, which uses the pricing scheme of (Deng et al., 2014). In (Deng et al., 2014), the peak demand at NE is reduced since prices are proportional to $\sum_{n=1}^N \boldsymbol{x}_{n,t}$. However, there is no guarantee on the total energy consumption which leads to an inefficient NE with high prices due to the high demand. Using our algorithm, the prices of producing and consuming electricity at NE drop dramatically, increasing the social welfare significantly.

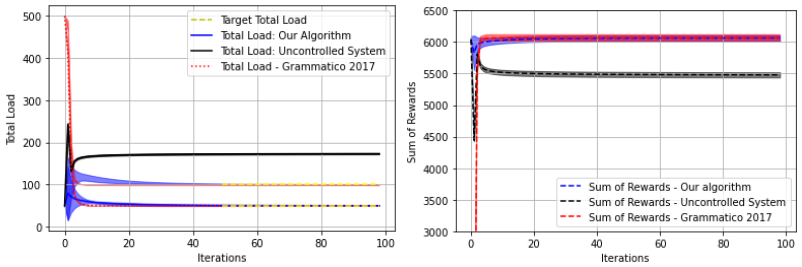
In Fig. 2c, we compare our algorithm to that of (Grammatico, 2017), which assumes utility functions of the form $u_n(\boldsymbol{x}) = f_n(\boldsymbol{x}_n) - \left(C\sum_{n=1}^N \boldsymbol{x}_n\right)^T \boldsymbol{x}_n - (K_g\boldsymbol{\lambda})^T \boldsymbol{x}_n$ for a strongly concave f_n, control vector λ, a designed matrix K_g and a matrix C which is assumed to be known to the manager. We simulated $N = 1000$ players with $K = 2$ and $f_n(\boldsymbol{x}_n) = \sum_{k=1}^K q_n^k\sqrt{1 + x_n^k}$ and $C = \frac{1}{100}$, where $\left\{q_n^k\right\}$ were chosen independently and uniformly at random on $[1, 5]$ and $K_g = \frac{1}{10}$. The action set was $\mathcal{X}_n = \left\{\boldsymbol{x}_n \,\middle|\, 0 \le x_n^k \le 1, \sum_{k=1}^K x_n^k \le 1, \forall k\right\}$ for each n. The target loads were $\boldsymbol{l}^* = \{50, 100\}$. We used $\eta_0 = 0.2$ and $\varepsilon_0 = 0.01$. The convergence time and the resulting social welfare of both algorithms are comparable, where (Grammatico, 2017) converges slightly faster but significantly overshoots the total loads. The algorithm in (Grammatico, 2017) assumes that the manager can replace the $\sum_{n=1}^N \boldsymbol{x}_{n,t}$ that affects the players through $u_n(\boldsymbol{x})$ with an arbitrary control signal vector σ. This assumption is impossible in scenarios where $\sum_{n=1}^N \boldsymbol{x}_{n,t}$ is indirectly measured by the players, as is often the case in DSM or power control. Hence, our algorithm is competitive with (Grammatico, 2017) while lifting this limiting assumption, requiring no knowledge of reward parameters, and applying to a more general class of games.



(a) Power control



(b) Demand-side management



(c) Comparison with (Grammatico, 2017)

Figure 2: Numerical performance. In (a) and (b), line formats indicate the method (left legend) and colors indicate different resources (right legend).

8. Conclusion

We studied how to balance loads in strongly monotone resource allocation games where the game structure is unknown to the manager. Our algorithm only requires the manager to observe the total load on each resource. We prove that our simple online algorithm guarantees the total load constraints $\boldsymbol{l}^*$ by adjusting the prices per unit of each resource $\boldsymbol{\alpha}$ in real-time. Specifically, the algorithm converges to a NE such that $\alpha^k = 0$ if the total load is less than l_k^* at equilibrium, and otherwise, the load is exactly l_k^*. The algorithm does not wait for the players to converge to a NE but uses the feedback from every turn, overcoming the associated "equilibrium noise". This is an encouraging result since our algorithm is easy to implement and maintains the users' privacy since the manager does not observe their individual actions, and they do not need to reveal their reward functions, which might even be unknown to them.

From a technical point of view, we analyze how the NE of the game changes when the manager changes $\boldsymbol{\alpha}_t$. Our algorithm essentially controls the unknown game such that its NE becomes desirable from an optimization point of view. The tools developed here to analyze the equilibrium noise can be useful to control unknown games beyond the case of resource allocation. Since convergence to NE is relatively fast, this has the potential to accelerate general cooperative multi-agent optimization (Bistritz & Bambos, 2020).

Controlling unknown games with bandit feedback is a new online learning paradigm. The game to be controlled can be thought of as a non-stationary but highly structured bandit. Our numerical simulations show fast convergence to the load-balanced NE, so it is interesting to analyze the convergence time, for which Lemma 5 gives the first clue. More generally, proving regret bounds on the manager's learning is an exciting new research avenue that can shed light on the convergence time of game control schemes.

References

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

Alatur, P., Levy, K. Y., and Krause, A. Multi-player bandits: The adversarial case. *Journal of Machine Learning Research*, 21(77):1–23, 2020.

Alpcan, T. and Pavel, L. Nash equilibrium design and optimization. In *2009 International Conference on Game Theory for Networks*, pp. 164–170. IEEE, 2009.

Alpcan, T., Başar, T., Srikant, R., and Altman, E. CDMA uplink power control as a noncooperative game. *Wireless Networks*, 8(6):659–670, 2002.

Alpcan, T., Pavel, L., and Stefanovic, N. A control theoretic approach to noncooperative game design. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 8575–8580. IEEE, 2009.

Alpcan, T., Nekouei, E., Nair, G. N., and Evans, R. J. An information analysis of iterative algorithms for network utility maximization and strategic games. *IEEE Transactions on Control of Network Systems*, 6(1):151–162, 2018.

Balcan, M.-F., Blum, A., Haghtalab, N., and Procaccia, A. D. Commitment without regrets: Online learning in Stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*, pp. 61–78, 2015.

Bambos, N. Toward power-sensitive network architectures in wireless communications: Concepts, issues, and design aspects. *IEEE Personal Communications*, 5(3):50–59, 1998.

Billingsley, P. *Probability and measure*. John Wiley & Sons, 2008.

Birmpas, G., Gan, J., Hollender, A., Marmolejo-Cossío, F. J., Rajgopal, N., and Voudouris, A. A. Optimally deceiving a learning leader in Stackelberg games. *arXiv preprint arXiv:2006.06566*, 2020.

Bistritz, I. and Bambos, N. Cooperative multi-player bandit optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

Bistritz, I. and Leshem, A. Distributed multi-player bandits-a game of thrones approach. In *Advances in Neural Information Processing Systems*, pp. 7222–7232, 2018.

Bistritz, I., Baharav, T., Leshem, A., and Bambos, N. My fair bandit: Distributed learning of max-min fairness with multi-player bandits. In *International Conference on Machine Learning*, pp. 930–940. PMLR, 2020.

Bourke, T. *Server load balancing*. " O'Reilly Media, Inc.", 2001.

Boursier, E. and Perchet, V. SIC-MMAB: synchronisation involves communication in multiplayer multi-armed bandits. In *Advances in Neural Information Processing Systems*, pp. 12048–12057, 2019.

Brown, P. N. and Marden, J. R. Optimal mechanisms for robust coordination in congestion games. *IEEE Transactions on Automatic Control*, 63(8):2437–2448, 2017.

Candogan, U. O., Menache, I., Ozdaglar, A., and Parrilo, P. A. Near-optimal power control in wireless networks: A potential game approach. In *2010 Proceedings IEEE INFOCOM*, pp. 1–9. IEEE, 2010.

Chavali, P., Yang, P., and Nehorai, A. A distributed algorithm of appliance scheduling for home energy management system. *IEEE Transactions on Smart Grid*, 5(1): 282–290, 2014.

Chen, H., Li, Y., Louie, R. H., and Vucetic, B. Autonomous demand side management based on energy consumption scheduling and instantaneous load billing: An aggregative game approach. *IEEE transactions on Smart Grid*, 5(4):1744–1754, 2014.

Deng, R., Yang, Z., Chen, J., Asr, N. R., and Chow, M.-Y. Residential energy consumption scheduling: A coupled-constraint game approach. *IEEE Transactions on Smart Grid*, 5(3):1340–1350, 2014.

Deng, R., Yang, Z., Chow, M.-Y., and Chen, J. A survey on demand response in smart grids: Mathematical models and approaches. *IEEE Transactions on Industrial Informatics*, 11(3):570–582, 2015.

Deng, Y., Lahaie, S., and Mirrokni, V. Robust pricing in dynamic mechanism design. In *International Conference on Machine Learning*, pp. 2494–2503. PMLR, 2020.

Dowling, C., Fiez, T., Ratliff, L., and Zhang, B. Optimizing curbside parking resources subject to congestion constraints. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 5080–5085. IEEE, 2017.

Facchinei, F. and Pang, J.-S. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

Ferguson, B. L., Brown, P. N., and Marden, J. R. The effectiveness of subsidies and taxes in atomic congestion games. *IEEE Control Systems Letters*, 2021.

Fiez, T., Chasnov, B., and Ratliff, L. Implicit learning dynamics in Stackelberg games: Equilibria characterization, convergence analysis, and empirical study. In *International Conference on Machine Learning*, pp. 3133–3144. PMLR, 2020.

Galeotti, A., Golub, B., and Goyal, S. Targeting interventions in networks. *Econometrica*, 88(6):2445–2471, 2020.

Grammatico, S. Dynamic control of agents playing aggregative games with coupling constraints. *IEEE Transactions on Automatic Control*, 62(9):4537–4548, 2017.

Heydaribeni, N. and Anastasopoulos, A. Distributed mechanism design for multicast transmission. *arXiv preprint arXiv:1803.08111*, 2018.

Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

Leng, Y., Dong, X., Wu, J., and Pentland, A. Learning quadratic games on networks. In *International Conference on Machine Learning*, pp. 5820–5830. PMLR, 2020.

Lu, H., Balseiro, S., and Mirrokni, V. Dual mirror descent for online allocation problems. *arXiv preprint arXiv:2002.10421*, 2020.

Ma, J., Deng, J., Song, L., and Han, Z. Incentive mechanism for demand side management in smart grid using auction. *IEEE Transactions on Smart Grid*, 5(3):1379–1388, 2014.

Magesh, A. and Veeravalli, V. V. Multi-user MABs with user dependent rewards for uncoordinated spectrum access. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 969–972. IEEE, 2019.

Maharjan, S., Zhu, Q., Zhang, Y., Gjessing, S., and Basar, T. Dependable demand response management in the smart grid: A Stackelberg game approach. *IEEE Transactions on Smart Grid*, 4(1):120–132, 2013.

Marden, J. R. and Roughgarden, T. Generalized efficiency bounds in distributed resource allocation. *IEEE Transactions on Automatic Control*, 59(3):571–584, 2014.

Marden, J. R. and Wierman, A. Distributed welfare games. *Operations Research*, 61(1):155–168, 2013.

Marden, J. R., Young, H. P., Arslan, G., and Shamma, J. S. Payoff-based dynamics for multiplayer weakly acyclic games. *SIAM Journal on Control and Optimization*, 48 (1):373–396, 2009.

Mehrabian, A., Boursier, E., Kaufmann, E., and Perchet, V. A practical algorithm for multiplayer bandits when arm means vary among players. In *International Conference on Artificial Intelligence and Statistics*, pp. 1211–1221. PMLR, 2020.

Melo, E. Congestion pricing and learning in traffic network games. *Journal of Public Economic Theory*, 13(3):351–367, 2011.

Mertikopoulos, P. and Zhou, Z. Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, pp. 1–43, 2016.

Mguni, D., Jennings, J., Macua, S. V., Sison, E., Ceppi, S., and De Cote, E. M. Coordinating the crowd: Inducing desirable equilibria in non-cooperative systems. *arXiv preprint arXiv:1901.10923*, 2019.

Molzahn, D. K., Dörfler, F., Sandberg, H., Low, S. H., Chakrabarti, S., Baldick, R., and Lavaei, J. A survey of distributed optimization and control algorithms for electric power systems. *IEEE Transactions on Smart Grid*, 8 (6):2941–2962, 2017.

Nayyar, N., Kalathil, D., and Jain, R. On regret-optimal learning in decentralized multiplayer multiarmed bandits. *IEEE Transactions on Control of Network Systems*, 5(1):597–606, 2016.

Nedic, A. and Ozdaglar, A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Nekouei, E., Alpcan, T., and Chattopadhyay, D. Game-theoretic frameworks for demand response in electricity markets. *IEEE Transactions on Smart Grid*, 6(2):748–758, 2014.

Parise, F. and Ozdaglar, A. A variational inequality framework for network games: Existence, uniqueness, convergence and sensitivity analysis. *Games and Economic Behavior*, 114:47–82, 2019.

Parise, F. and Ozdaglar, A. E. Analysis and interventions in large network games. *Available at SSRN 3692826*, 2020.

Parkes, D. C., Singh, S. P., and Yanovsky, D. Approximately efficient online mechanism design. In *NIPS*, 2004.

Robbins, H. and Siegmund, D. A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing methods in statistics*, pp. 233–257. Elsevier, 1971.

Rosen, J. B. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica: Journal of the Econometric Society*, pp. 520–534, 1965.

Sandholm, W. H. Pigouvian pricing and stochastic evolutionary implementation. *Journal of Economic Theory*, 132(1):367–382, 2007.

Scutari, G., Facchinei, F., Pang, J.-S., and Palomar, D. P. Real and complex monotone communication games. *IEEE Transactions on Information Theory*, 60(7):4197–4231, 2014.

Tatarenko, T. and Garcia-Moreno, L. A game theoretic and control theoretic approach to incentive-based demand management in smart grids. In *22nd Mediterranean Conference on Control and Automation*, pp. 634–639. IEEE, 2014.

Tatarenko, T. and Kamgarpour, M. Learning Nash equilibria in monotone games. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 3104–3109. IEEE, 2019.

Zhou, Z., Mertikopoulos, P., Moustakas, A. L., Bambos, N., and Glynn, P. Robust power management via learning and game design. *Operations Research*, 2020.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.