2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Admission Control for Games with a Dynamic Set of Players

Ilai Bistritz*and Nicholas Bambos†

***Abstract*—We consider open games where players arrive according to a Poisson process with rate λ and stay in the game for an exponential random duration with rate μ. The game evolves in continuous time where each player n sets an exponential random clock and updates her action $a_n \in \{0, \ldots, K\}$ when it expires. The players take independent best-response actions that, uninterrupted, can converge to a Nash Equilibrium (NE). This models open multiagent systems such as wireless networks, cloud computing, and online marketplaces. When λ is small, the game spends most of the time in a (time-varying) equilibrium. This equilibrium exhibits predictable behavior and can have performance guarantees by design. However, when λ is too small, the system is under-utilized since not many players are in the game on average. Choosing the maximal λ that the game can support while still spending a target fraction $0 < \rho < 1$ of the time at equilibrium requires knowing the reward functions. To overcome that, we propose an online learning algorithm that the gamekeeper uses to adjust the probability θ to admit an incoming player. The gamekeeper only observes whether an action was changed, without observing the action or who played it. We prove that our algorithm learns, with probability 1, a θ^* such that the game is at equilibrium for at least ρ fraction of the time, and no more than $\rho + \varepsilon(\mu, \rho) < 1$, where we provide an analytic expression for $\varepsilon(\mu, \rho)$. Our algorithm is a black-box method to transfer performance guarantees of distributed protocols from closed systems to open systems.**

I. Introduction

In many practical systems, users or devices come and go, creating an "open network". A wireless device connects to the network, transmits, and then disconnects [1]–[3]. Passengers route their way to their destination and then leave the system [4]. Yet the optimization of such systems traditionally considers a static network with a fixed set of players. This static optimization often guarantees that the system will exhibit good performance at equilibrium. However, if the network is open, these performance guarantees may not be relevant.

Large-scale system optimization calls for distributed algorithms. It is infeasible to collect all network parameters in a central server, and then compute the optimal solution in real time. Such a centralized scheme also violates user privacy and creates a cyber-security vulnerability since one server controls millions of devices. In a distributed algorithm, players take independent actions based on their local observations. A distributed system can be modeled as a game, where the reward function of a player guides her action choices. In this paper, we focus on best-response dynamics (BRD) as our distributed protocol, where a player that updates her action picks the best action given the current action profile and her reward function. For many classes of games [5], [6], a static set of players playing BRD converge to a Nash equilibrium (NE) [7]. With a dynamic set of players, every incoming player disrupts the convergence to this equilibrium.

If the arrivals are slow enough and players stay in the game for long enough, then the game will spend most of the time at equilibrium where it has static performance guarantees by design. However, with slow arrivals, the average number of players in the game is small which results in poor utilization. If the arrivals are too fast, then the game will always be far from equilibrium. An admission controller that decides whether to admit incoming players can balance between time at equilibrium and utilization. In a large-scale distributed system, the admission controller has little knowledge about the game and limited observations of the dynamics.

Formally, our objective is to find the maximal arrival rate λ^* that the system can support under the constraint that it spends at least a target fraction ρ of the time at equilibrium. Then the admission controller can accept each player with probability $\theta^* = \frac{\lambda^*}{\lambda}$, independently between players. However, calculating λ^* offline requires the manager to know the reward functions of the players and to observe their actions. These requirements are unrealistic for large-scale networks.

To overcome this challenge, we propose a simple online algorithm that can learn $\theta^* = \frac{\lambda^*}{\lambda}$ (even if it is non-unique). Motivated by large-scale networks, we only require that the admission controller (gamekeeper) can observe changes in the action profiles, arrivals, or departures. Our algorithm does not need to know the reward functions or observe the actions, or even who changed the action profile or how. Observing changes in the action profile is possible when the manager can sense the interference in a wireless network, server loads in cloud computing [8], or the total energy consumption. Therefore, our algorithm is "plug and play" with distributed protocols for many applications, converting their static performance guarantees to the open network scenario.

A. Related Work

Admission control is one of the main tasks in networking [1], [2], [4], [9], and learning algorithms for admission control have been studied in [10], [11]. In [12], the load balancing of server systems with an open network was studied. Efficient load-oblivious distributed protocols are designed, that

*Department of Industrial Engineering and School of Electrical Engineering, Tel Aviv University, Israel ilaibistritz@tauex.tau.ac.il
†Department of Electrical Engineering and Department of Management Science and Engineering, Stanford University, CA, USA bambos@stanford.edu

are shown to stabilize the system if this is feasible. Our approach here is different since we design an admission control algorithm for a given distributed protocol of the agents. This distributed protocol is designed with static performance (i.e., performance at equilibrium) in mind. Our motivation is to show that the static performance guarantees of these distributed protocols can be leveraged into the open network case in a black-box manner. Our work is the first to introduce online learning for admission control to games, to ensure that the game is at equilibrium for a target fraction of the time.

Game theory offers an analytical framework to design and analyze distributed protocols [13], [14]. Traditionally, game theory is used to predict the outcome of the interaction between selfish agents, via equilibria analysis. Then, mechanism design builds protocols that have performance guarantees that are robust to selfish behavior [15]. More recently, game theory has been used to design distributed protocols between programmed cooperative agents [16]–[22]. The latter is the case for wireless networks, autonomous vehicles, or a team of robots where all players are manufactured by the same company, or follow a standard. With cooperative agents, the reward functions do not model the players' selfish objectives but are designed to have a globally efficient NE.

Our work is also related to the literature on game control [23]–[29], where a manager adjusts reward parameters online (e.g., prices of resources) to achieve a global objective. However, our control of the game is only through the admission of players without affecting their reward functions.

Distributed systems with a dynamic set of players are also called "open multiagent systems" [30]–[33], and were studied in a game-theoretic context [34], [35]. In this paper, we study the control of such open multiagent systems. Specifically, we are interested in admission control that guarantees that the game spends a target fraction of the time at equilibrium.

II. Problem Formulation

Consider a game with a time-varying set of players (i.e., an open game). There is an admission controller, or a "gamekeeper", that decides if an arriving player is admitted into the game. Admitting too many players causes the game to spend a lot of time not at equilibrium, where no performance guarantees exist and behavior is unpredictable. Admitting too few players results in an under-utilized system. The gamekeeper optimizes the trade-off between user performance guarantees and utilization by controlling the fraction of time that the game is at equilibrium. The gamekeeper's goal is to guarantee that the game is at equilibrium for a $0 < \rho < 1$ fraction of the time by adjusting the admission probability θ.

Given some knowledge of the game, the designer can calculate offline a ρ that optimizes an objective. Without such knowledge, a wrapper algorithm can explore reasonable (e.g., high) values for ρ. In general, optimizing ρ is domain-specific and leverages the structure of the application. Nevertheless, in this paper, we assume ρ is an input to our algorithm.

A. The Open Game

Our game evolves in continuous time. At time t, the set of players in the game is $\mathcal{N}_t = \{1, \ldots, N_t\}$. Each player has a type $\nu \in \{0, \ldots, V\}$ for a finite V. Each of the players in the game at time t, $n \in \mathcal{N}_t$, plays an action $a_{n,t} \in \{0, 1, \ldots, K\}$. Players switch their actions asynchronously when their random clock expires, as we explain below. We assume that the number of players in the game is bounded by some finite capacity $N_{\max}$. This capacity is typically a function of the infrastructure and the physical limitations of the system. However, since our goal is to propose an admission control, maintaining that no more than $N_{\max}$ players are in the game can be viewed as part of the design and not as a modeling assumption.

To formalize a game with a dynamic set of players, we assume that the game always has $N_{\max}$ players, but some of them are dummy players (type $\nu = 0$) that represent empty player slots, so they are not "in the game". Then we can define the action profile at time t as $\boldsymbol{a}_t \in \{0, 1, \ldots, K\}^{N_{\max}}$. We define the reward function of player n from type ν as $u_n(\boldsymbol{a}) = u(\boldsymbol{a}; \nu_n)$, for some non-negative function u. We assume that $u(\boldsymbol{a}; 0) = 0$ for all $\boldsymbol{a}$, so dummy players obtain no rewards. Without loss of generality, we assume that dummy players play $a_n = 0$ so they do not affect other players.

We assume that the random arrivals of players follow a Poisson process with rate λ. Each arriving player has a random type $\nu_n = \nu$ with probability $p_\nu > 0$ for all $\nu \neq 0$, independently between arrivals, and this player's index is the minimal n that had $\nu_n = 0$ before the arrival (i.e., "empty player slot"). An admitted player ($\nu_n \neq 0$) plays $a_n = 0$ before her first action update since only after admittance she can deduce her best-response. An admitted player ($\nu_n \neq 0$) departs the game after a random duration that is exponentially distributed with rate μ. The arrival and departure processes of different players are all independent of each other.

The players in the game act asynchronously. To model this, we assume that each player n inside the game has a random exponential clock c_n with rate 1, which is independent of all the other clocks. When c_n expires at time t, player n updates her action using best-response dynamics (BRD):

$$a_{n,t} \in \underset{a_n \in \{1,\ldots,K\}}{\arg\max}\; u_n\left(a_n, \boldsymbol{a}_{-n,t}\right) \tag{1}$$

where ties are arbitrarily broken. An action profile $\left(a_n^*, \boldsymbol{a}_{-n}^*\right)$ is a NE at time t if $a_{n,t} = \underset{a_n \in \{1,\ldots,K\}}{\arg\max}\; u_n\left(a_n, \boldsymbol{a}_{-n,t}\right)$ for all $n \in \mathcal{N}_t$. We define $\mathcal{E}^*$ as the set of all $(\boldsymbol{a}, \boldsymbol{\nu})$ such that $\boldsymbol{a}$ is a NE given $\boldsymbol{\nu}$, which includes all possible NE of the open game, for all $\boldsymbol{\nu} \in \{0, 1, \ldots, V\}^{N_{\max}}$. We call an "unhappy" player that would switch an action given the opportunity to play a "deviator". Let D_t be the number of deviators at time t. Due to the exponential clocks, the decisions of players become more frequent when the system is more congested, as in practice.

Finally, we state our assumption on the open game.

Definition 1. We say that an open game is non-degenerate if for every n, $\boldsymbol{a}_{-n}$ and $\boldsymbol{\nu}$ such that $\nu_n \neq 0$, there exists an a_n' such that $u\left(\left(a_n', \boldsymbol{a}_{-n}\right); \nu_n\right) > u\left(\left(0, \boldsymbol{a}_{-n}\right); \nu_n\right) = 0$.

This assumption implies that every incoming player, of any type and in any action profile, has at least one action that is preferable to doing nothing. Without this assumption, players may have no reason to join the game. Then, the open game might degenerate in the sense that it stays at equilibrium forever despite the arrival of new players. With this assumption, any incoming player is a deviator until she updates her action from $a_n = 0$. In practice, it takes a while for a player to measure the environment and compute her best-response, which can only be done after the player is admitted (e.g., a drone gets permission to fly and observes the location of nearby drones). Technically, our analysis holds as is if any arrival increases D_t by at least one for whatever reason.

B. The Controlled Dynamics

The main challenge is that the gamekeeper does not know the reward functions of the players and cannot observe their actions. In large-scale systems, it is infeasible to collect data from all over the network, even if privacy is not an issue. The gamekeeper also cannot supervise the players' behavior, that are making independent decisions locally. This is the case with distributed cooperative agents or with selfish players that are unwilling to coordinate. We therefore only assume that the gamekeeper can observe if the action profile has changed, or if a player arrives or leaves the game. For example, if each action is a resource choice, this assumption would hold if the gamekeeper can monitor the loads on the resources. Based on this minimal feedback, the gamekeeper learns to control the admission probability θ such that asymptotically, the game will be at equilibrium a ρ fraction of the time, with small error.

Next, we define the action and type profile process:

Definition 2. Define the set of action profiles $\mathcal{A} = \{0, 1, \ldots, K\}^{N_{\max}}$ and the set of type profiles $\mathcal{V} = \{0, 1, \ldots, V\}^{N_{\max}}$. Define the random process in continuous time $\boldsymbol{X}_t(\theta) \triangleq (\boldsymbol{a}_t, \boldsymbol{\nu}_t) \in \mathcal{A} \times \mathcal{V}$, where $\theta \in [0, 1]$ is a fixed admission probability that the gamekeeper uses (see Algorithm 1). We also denote $\boldsymbol{x} = (\boldsymbol{a}, \boldsymbol{\nu})$ and $\mathcal{X} = \mathcal{A} \times \mathcal{V}$ for shorthand.

The process $\boldsymbol{X}_t(\theta)$ can be thought of as the union of all the absorbing Markov chains of the "closed" games (one for each type profile), connected through the perturbations of arrivals and departures. This is formalized next and proved in [36]:

Proposition 3. *If $\theta > 0$, then $\boldsymbol{X}_t(\theta)$ is a continuous time ergodic Markov chain with a unique stationary distribution $\boldsymbol{\pi}(\theta)$. Let $\boldsymbol{X}_l(\theta)$ be the jump (embedded) process of $\boldsymbol{X}_t(\theta)$ for integers $l \geq 0$. Then $\boldsymbol{X}_l(\theta)$ is an ergodic Markov chain with a unique stationary distribution $\boldsymbol{\varphi}(\theta)$.*

III. Online Learning Admission Control

In this section, we present our admission control algorithm (Algorithm 1) and its convergence guarantees (Theorem 4). The admission mechanism is simple - each arriving player is admitted at random with probability θ_l. The probability θ_l is updated every "jump", which occurs every time one out of three events occur: action switch, arrival, or departure. The

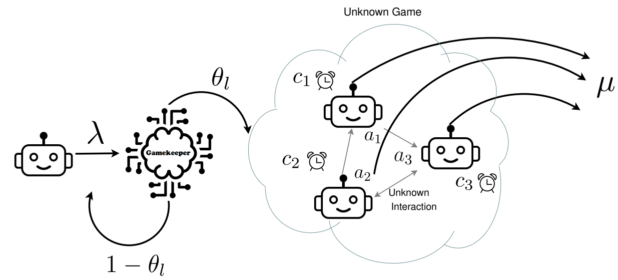


Fig. 1. Open game control

parameter θ^* that leads to the system being at equilibrium exactly ρ fraction of the time is unknown. Hence, the gamekeeper decreases θ_l if it thinks the game is not at equilibrium and increases θ_l otherwise. To this end, the gamekeeper estimates whether the game was at equilibrium solely based on s_l, which indicates if a player switched her action between the l-th jump at time t_l and the $l+1$-th jump at time t_{l+1}. This results in bias and noise that are unique to our open game control setting and require a novel convergence analysis [36]. The factor $t_{l+1} - t_l$ in (5) is the time spent in this action profile, and is necessary since we want the stationary distribution of the continuous-time Markov chain, $\boldsymbol{X}_t(\theta)$, to spend a ρ fraction of the time at equilibrium, and not that of $\boldsymbol{X}_l(\theta)$.

Our main result (proved in [36]) shows that by using Algorithm 1, the game will eventually be at equilibrium approximately a ρ fraction of the time, with an error $\varepsilon(\rho, \mu)$.

Theorem 4. *Let $\{\eta_l\}$ be the non-increasing step-size sequence of the gamekeeper such that $\sum_{l=0}^{\infty} \eta_l = \infty$ and $\sum_{l=0}^{\infty} \eta_l^2 < \infty$. Let $0 < \rho < 1$, which is an input to Algorithm 1. Define the stationary equilibrium probability as*

$$\pi_E(\theta) = \sum_{(\boldsymbol{a}, \boldsymbol{\nu}) \in \mathcal{E}^*} \pi_{(\boldsymbol{a}, \boldsymbol{\nu})}(\theta) \tag{2}$$

where $\mathcal{E}^*$ is the set of all NE and $\boldsymbol{\pi}(\theta)$ is the stationary distribution of $\boldsymbol{X}_t(\theta) = (\boldsymbol{a}_t, \boldsymbol{\nu}_t)$. Assume that $\pi_E(1) < \rho$. Assume a non-degenerate open game (Definition 1). Then, θ_l in Algorithm 1 converges with probability 1 as $l \to \infty$ to a set Θ^* of θ^* such that $\rho \leq \pi_E(\theta^*) \leq \rho + \varepsilon(\rho, \mu)$, where

$$\frac{\varepsilon(\rho, \mu)}{1 - \rho} = \max_{0 \leq \theta \leq \theta_H} \min_{n^*} \frac{\frac{\sum_{n=n^*+1}^{N_{\max}} \frac{\left(\frac{\lambda\theta}{\mu}\right)^n}{n!}}{\lambda\theta \sum_{n=0}^{N_{\max}-1} \frac{\left(\frac{\lambda\theta}{\mu}\right)^n}{n!}} + \mu n^* + \lambda\theta}{1 + \mu n^* + \lambda\theta} \tag{3}$$

and $0 < \theta_H \leq 1$ is calculated in Algorithm 1.

Theorem 4 argues about the stationary distribution of $\boldsymbol{X}_t(\theta_l)$ when θ_l has converged. Thus, the fraction of time spent at equilibria will approximate ρ asymptotically in time. We show numerically fast convergence in Section IV. Nevertheless, an open system is a long-term service with no time limit (e.g., cloud computing, wireless networks), so its steady state behavior is more important than its convergence time.

The error $\varepsilon(\rho, \mu)$ stems from the learning: the gamekeeper has to estimate if the game is at equilibrium based only on action deviations as feedback. This estimation is biased, and the bias depends on the stationary equilibrium probability and the estimation error which depend on the game (i.e., reward functions). Therefore, as long as the game is unknown, the algorithm cannot zero this bias, which results in $\varepsilon(\rho,\mu)>0$.

The error bound $\varepsilon(\rho,\mu)$ is also necessary to calibrate Algorithm 1 that cannot guarantee otherwise that at least a ρ fraction of the time will be spent at equilibrium since it overestimates the time the game spends at equilibrium.

We assume that $\pi_{\mathrm{E}}(1)<\rho$ since otherwise we cannot admit more players to reduce $\pi_{\mathrm{E}}(\theta)$ below $\rho+\varepsilon(\rho,\mu)$ with their perturbations. Nevertheless, if $\pi_{\mathrm{E}}(1)\geq\rho$ then no admission control is needed since $\theta^*=1$ is optimal: all arriving players are accepted and the game still spends more than ρ fraction of the time at equilibrium. Algorithm 1 does not need to know if $\pi_{\mathrm{E}}(1)<\rho$, and will converge to $\theta^*=1$ if $\pi_{\mathrm{E}}(1)>\rho$.

Remark 5 (**Monotone** $\pi_E(\theta)$). In non-pathological scenarios, the stationary probability of being at equilibrium $\pi_{\mathrm{E}}(\theta)$ is decreasing in θ since incoming players perturb the system, so the faster the arrivals are the less time the game is at equilibrium. Then, Theorem 4 implies that Algorithm 1 converges to the unique θ^* such that $\pi_{\mathrm{E}}(\theta^*)=\rho$ up to the error $\varepsilon(\rho,\mu)$. This θ^* is the maximal admission probability such that $\pi_{\mathrm{E}}(\theta^*)\geq\rho$, so it maximizes the average number of players in the game given the time at equilibrium constraint. However, to keep our result general, we do not assume that $\pi_{\mathrm{E}}(\theta)$ is decreasing so θ^* does not have to be unique.

Remark 6 (**Equilibrium Existence**). Assuming that a NE exists is not technically needed since an empty game is trivially at equilibrium. In the extreme case where the game has no NE other than when it is empty, the algorithm will force the game to be empty around a fraction ρ of the time. When equilibria exist for most type profiles, and the dynamics converge to these equilibria, the algorithm will not force the game to be empty a fraction ρ of the time, because then together with the non-trivial equilibria time, the fraction of time at equilibrium will exceed $\rho+\varepsilon(\rho,\mu)$. In fact, if $\pi_{\mathrm{E}}(\theta)$ is decreasing in θ, the algorithm will let as many players in as possible such that the fraction of time at equilibrium is approximately ρ. Therefore, the faster convergence to equilibrium is, the higher the admission rate θ^* will be. With cooperative distributed players, the designer can ensure that the dynamics converge to an existing equilibrium for most type profiles $\boldsymbol{\nu}\in\mathcal{V}$.

Algorithm 1 Online Open Game Control

Initialization: Let $(\boldsymbol{a}_0,\boldsymbol{\nu}_0)\in\mathcal{X}$. Set $l=0$ and $t_l=0$ for all integers $l\geq 0$. Let $\{\eta_l\}$ be a non-increasing positive sequence such that $\sum_{l=0}^{\infty}\eta_l=\infty$ and $\sum_{l=0}^{\infty}\eta_l^2<\infty$. Set $\theta_0=1$.
Input: $0\leq\rho\leq 1$, λ, μ, $N_{\max}$.
Calibration:

1) Compute $0<\theta_H\leq 1$ such that $\pi_{\mathrm{E}}(\theta_H)<\rho$ using $\pi_{\mathrm{E}}(\theta)\leq 1-\left(1-\frac{\frac{\left(\frac{\lambda\theta}{\mu}\right)^{N_{\max}}}{N_{\max}!}}{\sum_{n=0}^{N_{\max}}\frac{\left(\frac{\lambda\theta}{\mu}\right)^n}{n!}}\right)\lambda\theta.$
2) Compute $\tilde{\rho}\triangleq\rho+\varepsilon(\rho,\mu)$ for $\varepsilon(\rho,\mu)$ as defined in (3).
3) Compute $\theta_L>0$ such that $\pi_{\mathrm{E}}(\theta_L)>\tilde{\rho}$ using $\pi_{\mathrm{E}}(\theta)\geq\frac{1}{\sum_{n=0}^{N_{\max}}\frac{\left(\frac{\lambda\theta}{\mu}\right)^n}{n!}}$.

For all continuous time $t\geq 0$:

1) **If the clock c_n expires for some player n:**
   a) Player n selects a new action a_n^{U} according to
$$a_n^{\mathrm{U}}\in\underset{a_n\in\{1,\ldots,K\}}{\arg\max}\ u_n(a_n,\boldsymbol{a}_{-n,t}). \qquad (4)$$
   b) If $\left(a_n^{\mathrm{U}},\boldsymbol{a}_{-n,t}\right)\neq\boldsymbol{a}_t$, then set $t_{l+1}=t$ and $s_l=1$ (switch detected).
2) **If a new player arrives**:
   a) If $N_t<N_{\max}$ then admit the player with probability θ_l, and reject otherwise.
   b) If $N_t=N_{\max}$ then: with probability θ_l, discard an existing player at random and admit the arriving player, and reject otherwise.
   c) If the player was admitted: set $t_{l+1}=t$ and $s_l=0$.
3) **If a player leaves**: set $t_{l+1}=t$ and $s_l=0$.
4) If $t_{l+1}>0$ (i.e., the $l+1$-th jump event has occurred) then update the admission probability
$$\theta_{l+1}=\Pi_{[\theta_L,\theta_H]}\left(\theta_l+\eta_l\left(t_{l+1}-t_l\right)\left(1-s_l-\tilde{\rho}\right)\right) \qquad (5)$$
   and then update $l\leftarrow l+1$, where $\Pi_{[\theta_L,\theta_H]}$ is the Euclidean projection into $[\theta_L,\theta_H]$.

End

IV. Numerical Simulations

Consider players that each is a wireless link - a pair of a transmitter and a receiver, such that transmitter n uses the transmission power P_n. The signal to noise and interference ratio (SINR) at link n's receiver is defined as $\Gamma_n(\boldsymbol{P})=\frac{g_{n,n}P_n}{N_0+\sum_m g_{m,n}P_m}$, where $g_{m,n}$ is the channel gain between transmitter m and receiver n, and N_0 is the channel noise variance. The type ν_n of player n is the location of transmitter n and receiver n on a 2D disk with radius r, as specified in the figures. The transmitter's location was chosen uniformly at random on the disk, and its transmitter location was chosen uniformly at random within radius r_n of the receiver, where r_n was chosen uniformly at random on $[1,1.1]$ independently between links. The locations were stored as floating points (i.e., discrete with high resolution). The channel gains were chosen as $g_{m,n}=\min\left\{\|\boldsymbol{x}_{T,m}-\boldsymbol{x}_{R,n}\|^{-2},1\right\}$ where $\boldsymbol{x}_{T,m}$ is the location of transmitter m and $\boldsymbol{x}_{R,n}$ is the location of receiver n. Hence, the channel gains $\{g_{n,m}\}$ are a function of the types. Let $\Delta>0$ and let $P_{\max}=K\Delta$ be the maximal transmission power possible. The action set of player n is
$$\mathcal{A}_n=\{P_{\max}\}\cup\{i\Delta\mid i=0,\ldots,K,\ \Gamma_n(i\Delta,\boldsymbol{P}_{-n})\geq\gamma_n^*\}$$
where γ_n^* is the target SINR chosen in advance. Hence, the action set is constrained by the actions of others, given by $\boldsymbol{P}_{-n}$. However, player n does not need to know $\boldsymbol{P}_{-n}$, since it only affects $\Gamma_n(\boldsymbol{P})$ through the interference $\sum_m g_{m,n}P_m$,

which player n can measure. In all experiments, we used $\gamma_n^* = 10$ for all n with $\Delta = 10^{-5}$ and $K = 10^5$. There was no noticeable difference in the results for a finer Δ.

The reward function of player n is $u_n(\boldsymbol{P}) = P_{\max} - P_n$, so players pick the minimal transmission power that satisfies their SINR constraint. This power control setting is a game, where the actions of one player affect the reward of others. This scenario is the discretized version of the one in [3]. In our open game case, when a link tries to connect to the network, the gamekeeper can admit or reject it. In a large-scale network, the reward functions and actions are unknown to the gamekeeper since it does not have the time or capacity to collect all locations (types) and transmission powers.

Performance was averaged over 1000 realizations. The shaded blue region includes one standard deviation below and above the average. In each realization, θ_0 was chosen uniformly at random on $[0, 1]$. The initial action of an admitted player was $P_n = 0$. We used $\tilde{\rho} = \rho + 0.25(1 - \rho)$ which assumes $\varepsilon(\rho, \mu) \leq 0.25(1 - \rho)$. We simulated the most difficult case of $\theta_L = 0, \theta_H = 1$. The x-axis is time in "seconds", and we used $\lambda = 1$, so a "second" is the average arrival rate. The gamekeeper's step size sequence was $\eta_l = \frac{0.3}{l^{0.7}}$.

Fig. 2 shows the convergence of our algorithm for two different scenarios. Our algorithm converges fast, since from around $t = 10^4$ on the fraction of time at equilibrium is close to ρ. The bias, which is the gap between where the blue curve converges to and the target ρ shown in red, at $t = 2 \cdot 10^4$, is ~1.24% in Fig. 2(a) and ~3.78% in Fig. 2(b), which are both better than the worst case bound of Theorem 4.

Fig. 3 shows the minimal reward and the sum of rewards of our algorithm compared to an alternative admission control. Allowing too many players in increases the interference which leads to large interference and zero reward for all. The alternative admission control uses no gamekeeper but adjusts $N_{\max}$ to match the average number of players in the game with the gamekeeper, which was ~ 8. This represents classic admission control that controls the number of players in the game directly. We see that even though both algorithms maintain the same number of players on average, the gamekeeper achieves both a better sum of rewards and a better minimal reward. This demonstrates that the improvement in performance by the gamekeeper is not only due to limiting the number of players but also because it maintains equilibrium for the vast majority of the time. The results in [3] imply that best-response dynamics converge to a NE. Interestingly, this NE is optimal in the sense that it minimizes the total transmission power under the SINR constraints if these are feasible. Hence, the fraction of time the system is optimal for the players inside the game is lower bounded by the fraction of time the game spends in a NE, as guaranteed by our algorithm.

V. Conclusions

While most distributed protocols are designed for a static set of agents, in many applications the set of agents is dynamic. Examples include wireless networks, cloud computing, transportation, and online marketplaces. A distributed protocol is typically designed such that the agents converge to an efficient equilibrium. This paper was motivated by converting the static performance guarantees of such protocols at equilibrium to this dynamic setting. This is achieved by designing a novel online admission control algorithm that treats the original distributed protocol, modeled as a game, as a black box. The algorithm adjusts online the probability θ_l to admit an incoming player. Our proposed algorithm only requires the manager (gamekeeper) to observe action switches, or if a player arrived or left the game. The gamekeeper does not need to know the reward functions of the players or observe their actions. We proved that our algorithm guarantees that the game is at equilibrium at least a target fraction ρ of the time and no more than $\rho + \varepsilon(\rho, \mu)$, providing a bound for $\varepsilon(\rho, \mu)$. The parameter ρ is set by the gamekeeper to optimize the tradeoff between user performance and system utilization.

Our work introduces the concept of controlling the admission of open games, which brings together game theory and queuing theory. The model presented here is only the first step towards unleashing the potential of these techniques. Extending our results beyond best-response dynamics would make them more broadly applicable.

References

[1] M. Naghshineh and M. Schwartz, "Distributed call admission control in mobile/wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 4, pp. 711–717, 1996.

[2] N. Bambos, S. C. Chen, and G. J. Pottie, "Channel access algorithms with active link protection for wireless communication networks with power control," *IEEE/ACM Transactions On Networking*, vol. 8, no. 5, pp. 583–597, 2000.

[3] G. J. 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.

[4] A. Y. Lam, Y.-W. Leung, and X. Chu, "Autonomous-vehicle public transportation system: scheduling and admission control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1210–1226, 2016.

[5] A. Fabrikant, A. D. Jaggard, and M. Schapira, "On the structure of weakly acyclic games," in *Algorithmic Game Theory*, Springer, 2010.

[6] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[7] J. Nash, "Non-cooperative games," *The Annals of Mathematics*, vol. 54, pp. 286–295, 1951.

[8] K. Schubert, N. Master, Z. Zhou, and N. Bambos, "Asynchronous best-response dynamics for resource allocation games in cloud computing," in *American Control Conference (ACC), 2017*, 2017.

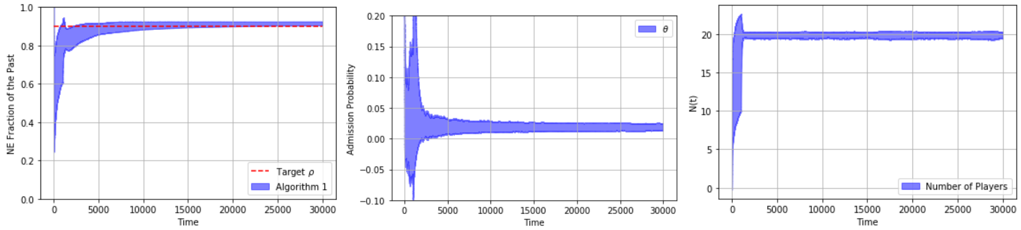
[9] S. C. Chen, N. Bambos, and G. J. Pottie, "Admission control schemes for wireless communication networks with adjustable transmitter powers," in *Proceedings of INFOCOM'94 Conference on Computer Communications*, pp. 21–28, IEEE, 1994.

[10] J. Carlström, "Decomposition of reinforcement learning for admission control of self-similar call arrival processes," in *NIPS*, pp. 1033–1039, Citeseer, 2000.

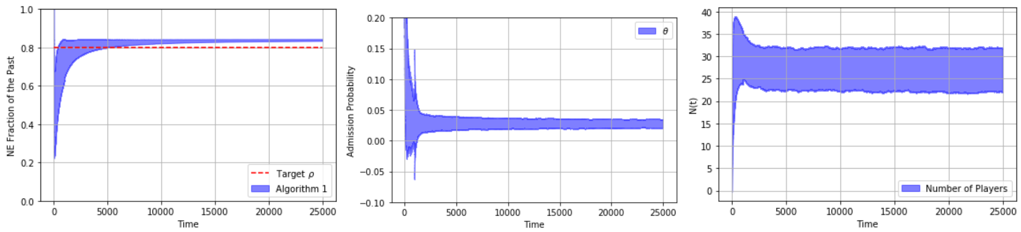
[11] T. X. Brown, H. Tong, and S. Singh, "Optimizing admission control while ensuring quality of service in multimedia networks via reinforcement learning," *Advances in Neural Information Processing Systems*, pp. 982–988, 1999.

[12] A. Ganesh, S. Lilienthal, D. Manjunath, A. Proutiere, and F. Simatos, "Load balancing via random local search in closed and open systems," *Queueing systems*, vol. 71, no. 3, pp. 321–345, 2012.

[13] J. Cohen, A. Héliou, and P. Mertikopoulos, "Learning with bandit feedback in potential games," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6372–6381, 2017.



(a) $r = \sqrt{10^4 N_{\max}}, \mu = 0.0001, \rho = 0.9, N_{\max} = 20$



(b) $r = \sqrt{10^4 N_{\max}}, \mu = 0.001, \rho = 0.8, N_{\max} = 40$

Fig. 2. Fraction of the time at equilibrium, admission probability, and number of players versus time

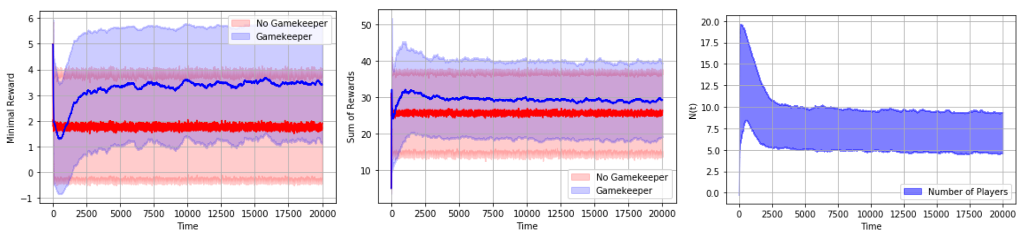


Fig. 3. Sum of rewards and minimal reward with and without the gamekeeper, $r = 10\sqrt{N_{\max}}$, $\mu = 0.001, \rho = 0.8, N_{\max} = 20$

[14] W. Krichene, B. Drighes, and A. Bayen, "On the convergence of no-regret learning in selfish routing," in *International Conference on Machine Learning*, pp. 163–171, PMLR, 2014.

[15] N. Heydaribeni and A. Anastasopoulos, "Distributed mechanism design for network resource allocation problems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 621–636, 2019.

[16] T. Alpcan, L. Pavel, and N. Stefanovic, "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*, 2009.

[17] E. Altman, A. Kumar, and Y. Hayel, "A potential game approach for uplink resource allocation in a multichannel wireless access network," in *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, 2009.

[18] J. R. Marden and A. Wierman, "Overcoming the limitations of utility design for multiagent systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1402–1415, 2013.

[19] J. R. Marden and T. Roughgarden, "Generalized efficiency bounds in distributed resource allocation," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 571–584, 2014.

[20] Z. Zhou, P. Mertikopoulos, A. L. Moustakas, N. Bambos, and P. Glynn, "Robust power management via learning and game design," *Operations Research*, vol. 69, no. 1, pp. 331–345, 2021.

[21] I. Bistritz and A. Leshem, "Game theoretic dynamic channel allocation for frequency-selective interference channels," *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 330–353, 2018.

[22] I. Bistritz, A. Ward, Z. Zhou, and N. Bambos, "Smart greedy distributed energy allocation: a random games approach," *IEEE Transactions on Automatic Control*, 2021.

[23] S. Grammatico, "Dynamic control of agents playing aggregative games with coupling constraints," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4537–4548, 2017.

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

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

[26] T. Alpcan and L. Pavel, "Nash equilibrium design and optimization," in *Game Theory for Networks, 2009. GameNets' 09. International Conference on*, 2009.

[27] I. Bistritz and N. Bambos, "Online learning for load balancing of unknown monotone resource allocation games," in *International Conference on Machine Learning*, pp. 968–979, PMLR, 2021.

[28] L. J. Ratliff and T. Fiez, "Adaptive incentive design," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3871–3878, 2020.

[29] S. Chandak, I. Bistritz, and N. Bambos, "Equilibrium bandits: Learning optimal equilibria of unknown dynamics," *arXiv preprint arXiv:2302.13653*, 2023.

[30] Y.-G. Hsieh, F. Iutzeler, J. Malick, and P. Mertikopoulos, "Optimization in open networks via dual averaging," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 514–520, IEEE, 2021.

[31] N. Hayashi, "Distributed subgradient method in open multiagent systems," *IEEE Transactions on Automatic Control*, 2022.

[32] M. Franceschelli and P. Frasca, "Stability of open multiagent systems and applications to dynamic consensus," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2326–2331, 2020.

[33] C. M. d. G. de Carnieres and J. M. Hendrickx, "Fundamental performance limitations for average consensus in open multi-agent systems," *IEEE Transactions on Automatic Control*, 2022.

[34] D. Shah and J. Shin, "Dynamics in congestion games," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 1, pp. 107–118, 2010.

[35] T. Lykouris, V. Syrgkanis, and É. Tardos, "Learning and efficiency in games with dynamic population," in *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 120–129, 2016.

[36] I. Bistritz and N. Bambos, "Gamekeeper: Online learning for admission control of networked open multiagent systems," 2023. tinyurl.com/GamekeeperExtended.