Distributed Reinforcement Learning based Delay Sensitive Decentralized Resource Scheduling

Geetha Chandrasekaran
Dept. of ECE, University of Texas at Austin

Gustavo de Veciana
Dept. of ECE, University of Texas at Austin

***Abstract*—We address the problem of distributed resource allocation in wireless systems in the presence of dynamic user traffic and coupling resulting from interference. We propose a Reinforcement Learning (RL) framework based on a separation of concerns between frequency reuse for interference mitigation and opportunistic user scheduling. In particular we explore a setting where a stochastic game is set up among base stations to learn frequency reuse patterns and solved using multi-agent RL given an underlying choice for user scheduling. We establish the existence and convergence to a Nash equilibrium of the proposed setting. The performance of our framework and theoretical findings are evaluated through simulation and compared to more aggressive *oracle-aided* centralized baselines. The resulting frequency reuse policy is shown to achieve 5–25% improvements in capacity and associated delay performance over a centralized interference aware max weight scheduling policy across BSs. Furthermore, a reduced physical resource utilization on the order of 9-34% leads to a higher energy efficiency as compared to the centralized benchmark.**

I. Introduction

WIRELESS networks have undergone tremendous change over the past decades going through various technology generations supporting higher data rates, improved network coverage and better user experience. A major factor enabling this progression has been network densification. While network densification improves users' data rates, it also may lead to reduced user traffic aggregation increasing the likelihood of bursty interference from neighboring base stations rendering static frequency reuse techniques less effective. This makes a dynamic frequency reuse scheme that can self tune to the network and traffic conditions highly desirable.

Resource allocation and power adaptation are problems in wireless systems where Reinforcement learning (RL) has proven to have some promise. Two key settings have been considered: cooperative Markov games and non-cooperative distributed games. Cooperative games typically draw on a more centralized decision making approach based on extensive information sharing posing practical limitations [1]. By contrast non-cooperative games typically involve distributed decision making based on local information typically leading to sub-optimal but more scalable approaches suitable to adapt to dynamic interference and user traffic.

A. Related work

Reinforcement Learning: There are many works proposing an RL approach to wireless resource allocation. For example, [2] propose a centralized learning system which periodically updates BS's policies (neural network model) giving it a partially centralized learning architecture. By contrast, we consider a completely distributed learning approach with variants that do not need any information exchange across BSs in the network. The work in [3], considers a heterogeneous network (HetNet) with macro and femto BSs sharing the spectrum in the same area and propose a Q-learning based algorithm for carrier selection and power allocation, but do not account for users' channel variability or interference from other HetNets. Most of the current literature based on a distributed RL approach to resource allocation, see [4]–[7], ignores the impact of user traffic dynamics and/or resource scheduling on the effective network throughput. In our work we show that user traffic dynamics can be learnt and devise resource allocation strategies that leverage this information.

RL algorithms have also been applied to mitigate interference in wireless networks. Techniques such as dynamic Q-learning [8], neural network [9], actor-critic RL [10], Deep Q-network (DQN) [11] have been used in a variety of settings such as HetNets, Cognitive radio and vehicular communication with the goal of resource allocation that either minimizes or mitigates interference. However, these articles fail to acknowledge and/or do not consider opportunistic scheduling, which we believe cannot be ignored when building a solution to distributed resource allocation.

Stochastic games and Scheduling: A centralized algorithm to resource allocation for interference mitigation and scheduling is considered in [12], but the approach results in increased computational complexity for larger networks. In contrast, we design autonomous learning at each BS that is scalable and of the same complexity irrespective of the network size. A downlink power control stochastic game between a macro BS and it's co-located small cell BSs has been considered in [13] without considering the impact of inter cell interference from other macro BSs. A resource allocation game among users using Code-Division Multiple Access (CDMA) system, with power and rate control for each user, has been considered in [14]. Note that interference from neighbouring CDMA networks has not been considered, this impacts the feasible range of power and transmit rate values. Distributed learning for resource selection and/or power allocation have also been considered in [15]–[18], but one or more of the following are not considered: interference from neighbouring BSs, user traffic dynamics and channel uncertainity due to time varying interference. For a more comprehensive survey of existing distributed learning approaches, the interested reader is directed to [19]. The vast majority of *scheduling policy* literature

on coupled queues [20]–[24] deal with the case where both queues can be serviced by the same server. Parallel queues with coupled service rates in the presence of channel aware scheduling has been discussed in [25], nevertheless cooperative strategies for interfering BSs has not been investigated.

B. Our Contributions

We consider the design of a resource allocation algorithm with dynamic user traffic for BSs coupled through interference. We propose a novel approach based on two coupled sub problems, which permits us to explore RL based interference mitigation having chosen a state-of-the-art (opportunistic) throughput optimal scheduler. This in turn reduces the state-space of the RL resource allocation problem leading to a quick training time (order of a couple of minutes in real time) and a resulting improvement in system capacity and performance. The main contributions of this work are as follows

1) We propose a systematic decomposition approach to optimizing frequency reuse under a *predetermined* dynamic user scheduling policy geared at making distributed RL techniques possible. To the best of our knowledge this has not been previously considered.
2) We propose and validate a proxy metric (reward) that enables distributed RL agents (base stations) to learn the interference-driven coupling amongst BSs.
3) Multiagent RL is a non-stationary stochastic game, existence and convergence to a Nash Equilibrium (NE) under specific conditions has been established in [26]. We show that our proposed algorithm satisfies these conditions, and hence show the existence and convergence to an NE.
4) We evaluate and compare through detailed simulation the potential of our proposed distributed approach vs an aggressive and more centralized baseline algorithms. Our results exhibit capacity gains of 5-25% over full frequency reuse as well as associated improvements in delay performance with an improved energy efficiency on the order of 9-34%.

II. System Model

A. Network Model

We consider downlink transmissions from a set of wireless BSs $\mathcal{B}$ of cardinality $|\mathcal{B}| = B$, serving a set of users/devices $\mathcal{U}$ such that $|\mathcal{U}| = U$, as shown in Fig. 1. The dynamics of the system evolve in discrete time, corresponding to transmission frames which are synchronized across BSs. Each frame consists of N Resource Blocks (RBs) each corresponding to a slice of sub carriers and slice of time (mini slot) within the frame. Each RB can be assigned by a BS b to serve at most one of its set of associated users $\mathcal{U}^b$. Let $A_u(t)$ be a random variable denoting the arrivals (in packets) for user u during time slot t and thus available for transmission at $t+1$. We shall assume that a user's arrivals across time slots are independent and identically distributed (i.i.d). Let $\lambda_u = E[A_u(t)]$ denote the mean packet arrivals per time slot for user u and let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_U)$. In the sequel $Q_u(t)$ will denote the queue length (in packets) of user u, i.e., the data available for transmission in time slot t and $\boldsymbol{Q}^b(t) = (Q_u(t) : u \in \mathcal{U}^b)$ the queues at BS b, while $\boldsymbol{Q}(t) = (\boldsymbol{Q}^b(t) : b \in \mathcal{B})$ the overall queue state of the system.

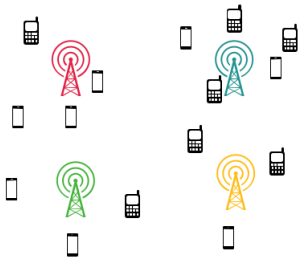

Fig. 1: Network Model

The channel gain between BS b and user u in slot t is modelled by a random variable $G_u^b(t)$ and assumed i.i.d across time. Let $\mathbf{G}(t) = (G_u^b(t) : b \in \mathcal{B}, u \in \mathcal{U})$ denote the gains amongst all BSs and users, while $\mathbf{G}^b(t) = (G_u^b(t) : u \in \mathcal{U}^b)$ denotes solely those between BS b and its associated users. For simplicity, we will assume flat fading i.e., the gains do not depend on the resource block/subcarriers, this can be easily generalized. We denote the mean channel gains by $\bar{g}_u^b = E[G_u^b(t)]$ along with associated vector notations $\bar{\boldsymbol{g}}^b$ and $\bar{\boldsymbol{g}}$. The resource allocation of RBs to users is modelled as a two step process. First a frequency reuse decision is made which determines the subset of RBs *available* for user allocation at each BS. A set of binary decisions are made at each base station b for an RB k on slot t: $S_k^b(t)$ is such that if $S_k^b(t) = 1$ if RB k is available for use by the BS, and if $S_k^b(t) = 0$ it is not to be used. Second, a scheduling decision is made determining which (if any) users are scheduled to transmit on the *available* subset of RBs. We let $\mathbf{S}^b(t) = (S_k^b(t) : k = 1, \ldots, N)$ denote the frequency reuse state of BS b at time t and $\mathbf{S}(t) = (\mathbf{S}^b(t) : b \in \mathcal{B})$ the overall frequency reuse state of the network. It will be convenient to let $\mathcal{S}^b(t) = \{k : S_k^b(t) = 1\}_{k=1,\ldots,N}$ denote the set of available RBs at base station b.

In general a scheduling policy $\mathbf{h}$ is an assignment h^b for each BS b of the available RBs to it's users. The assignment may depend on the available information denoted $\mathbf{I}^b(t)$, so a scheduling policy h^b for BS b is a mapping,

$$h^b(\,\cdot\,; \mathbf{I}^b(t)) : \mathcal{S}^b(t) \to \mathcal{U}^b \cup \{0\}\,, \tag{1}$$

assigning each RB made available by the frequency reuse policy $\mathcal{S}^b(t)$ to one of its users $\mathcal{U}^b$ or none at all, represented by user 0. Typically a BS scheduler will only have local information such as its users' channel gains and queue lengths, e.g., $\mathbf{I}^b(t) = (\mathbf{S}^b(t), \mathbf{G}^b(t), \mathbf{Q}^b(t))$. For simplicity we shall equivalently represent the result of scheduling via binary variables $\mathbf{h}(t) = (h_{uk}^b(t) : b \in \mathcal{B}, u \in U^b, k = 1, \ldots N)$ where $h_{uk}^b(t) = 1$ if the scheduler allocated an available RB $k \in \mathcal{S}^b(t)$ to user $u \in \mathcal{U}^b$ of BS b, otherwise it is 0.

In practice a BS's scheduler has access to Channel Quality Indicator (CQI) as well as estimates of previously observed interference and/or success/failure of transmissions for each

of its associated users, based on which it estimates the users' current Signal to Interference and Noise ratio (SINR). For simplicity we assume an adaptive modulation and coding scheme at the transmitter that can make use of this information to achieve a data rate close to the Shannon capacity. We understand that the Shannon capacity serves as a *rough* upper bound to the achievable rate, nevertheless, to keep things simple we use Shannon capacity as a rate metric to compare various algorithms in this work.

Due to possible interference from neighboring BSs, the transmission user rate under a given resource schedule is a complex function of all scheduled users. An idealized model might be as follows: if $h^b_{uk}(t) = 1$ the SINR for user u of BS b on RB k is

$$\text{SINR}^b_{uk}(t) = \frac{PG^b_u(t)}{\sum\limits_{b':b'\neq b}\sum\limits_{u'\in\mathcal{U}^{b'}} Ph^{b'}_{u'k}(t)G^{b'}_u(t) + N_0}, \quad (2)$$

where the numerator corresponds to received transmit power, and denominator the sum of intercell interference and noise. The downlink transmission rate to user $u \in \mathcal{U}_b$ on resource k at time t is given by,

$$c^b_{uk}(t) = n\mu\frac{W}{2}\log(1 + \text{SINR}^b_{uk}(t)) \text{ bits}, \quad (3)$$

where n is the number of subcarriers per RB and μ is time duration of an RB. Thus aggregating across the RBs of BS b we denote the total transmissions to user u in slot t as,

$$c_u(t) = \sum_{k=1}^{N} h^b_{uk}(t)c^b_{uk}(t) \text{ bits}\,, \quad (4)$$

where we suppressed the superscript b in $c^b_u(t)$ since each user is served by only one BS. Hence, under such a scheduling policy the queue dynamics for user u are given by

$$Q_u(t+1) = [Q_u(t) - f(c_u(t))]^+ + A_u(t)\,, \quad (5)$$

where $[x]^+ = \max[0, x]$ and $f(x)$ is an integer valued non-decreasing function on x modeling the packet departures at a user queue as a function of the SINR.

In the sequel we will find it useful to introduce the following notation. Note that given a frequency reuse state $\mathbf{S}(t) = \mathbf{s}$, a scheduler (1), channel and queue states $\mathbf{G}(t) = \boldsymbol{g}$, $\mathbf{Q}(t) = \boldsymbol{q}$, the service to user u can be written as,

$$c_u(t) = c_u(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q}). \quad (6)$$

Note that cellular networks can determine the *interference-free* signal to noise ratio (SNR) based on the user location through state of the art machine learning techniques [27]. For a given user $u \in \mathcal{U}^b$, the SNR at time t is denoted,

$$\text{SNR}_u(t) = \frac{PG^b_u(t)}{N_0}, \quad (7)$$

and a user's *effective "interference free" capacity* per RB for a channel strength $G^u_b(t) = g^u_b$ is denoted,

$$\kappa_u(g^u_b) = n\mu\frac{W}{2}\log\left(1 + \frac{Pg^b_u}{N_0}\right) \text{ bits}\,. \quad (8)$$

The design of a performance optimal frequency reuse and scheduling policy for this stochastic network system with queues which are coupled through interference is an exceedingly challenging problem. In this paper we propose a separation of concerns where the underlying BS schedulers are fixed, e.g., to a state-of-the-art opportunistic scheduler based on local information. Given the underlying scheduler we propose to have BSs learn how to manage frequency reuse so as to reduce the impact of inter-cell interference.

B. Markov Game: Learning frequency reuse policies

We formulate the problem of determining an overall frequency reuse policy across BSs as a Markov game [28] where each BS decides on it's own reuse policy so as to either (a) maximize its own reward, or (b) maximize a shared network reward. The rewards are a result of the BSs' frequency reuse decisions, and the underlying BS scheduling policies as well as the underlying environment/dynamics.

More formally, we consider a B-player Markov game

$$\langle \mathcal{S}^1, \ldots, \mathcal{S}^B; \mathcal{A}^1, \ldots, \mathcal{A}^B; p^1, \ldots, p^B; r^1, \ldots, r^B \rangle \quad (9)$$

including the following elements.

- $\mathcal{S}^b = \{0, 1\}^N$ denotes the set of possible frequency reuse states for for BS b i.e., values $\mathbf{S}^b(t)$ can take.
- $\mathcal{A}^b$ denotes the set of all possible actions BS b can take.
- $p^b(\boldsymbol{s}'^b|\boldsymbol{s}^b, \boldsymbol{a}^b)$ models the transition probabilities to the next state $\boldsymbol{s}' \in \mathcal{S}$, given that the current state and action pair given by $(\boldsymbol{s}^b, \boldsymbol{a}^b)$.
- $r^b(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q})$ corresponds to a reward associated with users scheduled at BS b on a given time slot conditional on the *overall* frequency reuse state $\boldsymbol{S}(t) = \boldsymbol{s}$, channel gains $\boldsymbol{G}(t) = \boldsymbol{g}$ and user queues $\boldsymbol{Q}(t) = \boldsymbol{q}$.

Below we describe several approaches to defining the rewards and action space for this game. Note that the frequency reuse game is such that BSs do not have access to the entire network state, in particular to the frequency reuse state, channels and queues of *other* BSs.

C. Actions and Rewards: Non cooperative Markov game

An action $\boldsymbol{a}^b \in \mathcal{A}^b$ determines the next frequency reuse state for BS b. We consider two possible models for the action space $\mathcal{A}^b$, $\{0, 1\}^N$ or $\{1, \ldots, N\}$, depending on the admissible action state complexity. When $\mathcal{A}^b = \{0, 1\}^N$, the frequency reuse state for the next frame is deterministically set to $\boldsymbol{s}'^b = \boldsymbol{a}^b$. In the second model where the action space is $\mathcal{A}^b = \{1, \ldots, N\}$, an action $a^b = k$ corresponds to a decision to transmit only on k RBs in the subsequent frame, with the RB positions chosen uniformly at random.

We design the per slot reward for each BS b to capture both the amount of data transmitted and the "efficiency" of such transmissions. In particular, given a frequency reuse state $\boldsymbol{s}$, and the scheduling decisions associated with channel gains $\boldsymbol{g}$ and queue lengths $\boldsymbol{q}$, the reward at BS b is modeled by,

$$r^b(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q}) = \sum_{u\in\mathcal{U}^b} \frac{c_u(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q})}{\kappa_u(\boldsymbol{g})}, \quad (10)$$

where $c_u(\cdot)$, defined earlier in (6) is the overall bits delivered to user u and κ_u defined in (8) is the *effective interference free capacity* of user u. This rewards the transmission of data to users at the BS, but penalizes transmissions experiencing excessive interference. Note that each agent in the Markov Game only sees its own frequency reuse state $\boldsymbol{s}^b$, whence it sees a reward $r^b((\boldsymbol{s}^b, \boldsymbol{s}^{-b}), \boldsymbol{g}, \boldsymbol{q})$ that depends on the frequency reuse actions of other players denoted $\boldsymbol{s}^{-b}$, the stationary distribution of the networks channel gains $\boldsymbol{G}(t)$ and possibly not stationary distribution of the network queues $\boldsymbol{Q}(t)$.

We consider a non cooperative Markov game where each BS learns a policy based on rewards either generated by its own users or all users in the network. The learned frequency reuse policy $\boldsymbol{\pi} \triangleq (\pi^b,\ b \in \mathcal{B})$ induces a set of transition probabilities on the frequency reuse states $(\mathcal{S}^b,\ b \in \mathcal{B})$ such that the expected long term rewards are maximized. We consider three different game settings based on the rewards and/or action space.

(G1) **Global reward game**: Each BS trains on the sum reward $\sum_{b\in\mathcal{B}} r^b(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q})$ generated by all BSs. Each BS b has an action space $\mathcal{A}^b = \{0, 1\}^N$.

(G2) **Local reward game**: Each BS trains on its own *local* reward $r^b(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q})$. Each BS has an action space $\mathcal{A}^b = \{0, 1\}^N$.

(G3) **Random action game**: Each BS trains on its own *local* reward $r^b(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q})$. Each BS has an action space $\mathcal{A}^b = \{1, \ldots, N\}$.

We can thus model the frequency reuse state transitions as a Markov chain induced on the frequency reuse state space by policy $\boldsymbol{\pi}$ and scheduling rule $\boldsymbol{h}$. With a slight abuse of notation, we use $(\pi^b(\boldsymbol{s}) : \boldsymbol{s} \in \mathcal{S}^b)$ to also denote the steady state distribution of the induced Markov chain at BS b. Note that the frequency reuse policy $\boldsymbol{\pi}$ in conjunction with a scheduling rule $\boldsymbol{h}$ determines the users' queue length distributions.

III. Proposed Solution: Probabilistic Frequency Reuse (PFR)

Given a traffic load $\boldsymbol{\lambda}$, one would like to pick a set of frequency reuse policies $\boldsymbol{\pi}$ from the set of all feasible policies $\mathcal{P}$ for interference mitigation and scheduler $\boldsymbol{h} \in \mathcal{H}$ that can either stabilize the user queues or maximize some network utility. Consider the B-player Markov game summarized in (9), we fix the scheduler $\boldsymbol{h}$ at each BS and learn interference management policies $\boldsymbol{\pi}$ using one of three game settings (G1), (G2) or (G3) based on our carefully chosen proxy metric (10).

We propose that each BS use an efficient algorithm to learn it's own frequency reuse policy in a distributed manner. Multi agent Q learning [26] is a model free learning algorithm for non cooperative Markov games. A schematic representation of our learning algorithm is depicted in Fig. 2. An agent at each BS b learns[1] its frequency reuse policy π^b using the rewards generated by the underlying BS scheduler. The scheduler then allocates RBs made available by the frequency reuse policy to users based on their channel quality and queue length. Finally the learning agent at each BS trains on the reward metric for each resource selection configuration $\boldsymbol{s}^b$ based on which the reuse policy is updated to maximize discounted future rewards. We will refer to this distributed method of learning as Probabilistic frequency reuse where each BS learns the fraction of time it should spend on a particular frequency reuse state to alleviate interference.

[1]After a random initialization of the reuse policy, the rewards generated by the scheduler is used to iteratively improve the policy.

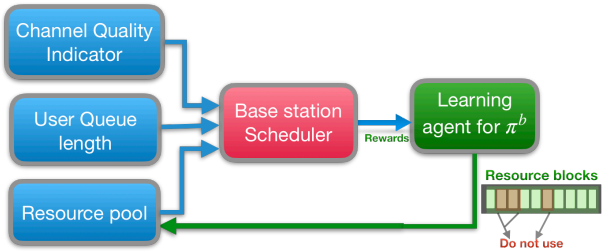

Fig. 2: Block diagram representation of our proposed system architecture at base station b.

Let $\boldsymbol{\pi}$ be the multi agent frequency reuse policy learnt by the distributed algorithm, then for a given initial state $\boldsymbol{s}^b$, the learning agent at each BS maximizes it's value function $v^{\pi^b}(\cdot)$

$$v^{\pi^b}(\boldsymbol{s}^b) = \sum_{k=1}^{\infty} \gamma^k \mathbb{E}^{\boldsymbol{\pi},\chi} \left[r^b(\mathbf{S}^b_{t+k}, \mathbf{G}_{t+k}, \mathbf{Q}_{t+k}) | \boldsymbol{\pi}, \mathbf{S}^b_t = \boldsymbol{s}^b \right]. \tag{11}$$

Note $r^b(\cdot)$ is the reward of BS b at time t and k is time step to capture future rewards. We start training with $\mathbf{Q}_0 = \mathbf{0}$ and after sufficiently long training time, the frequency reuse policy converges to a stationary distribution π^b which induces a distribution χ on the queue length. Also, each BS has access to the reward of other BSs either directly (as in global reward setting) or indirectly through the interference that each BS sees. Furthermore, the actions of each BS in the network are either indirectly observable at each BS through interference or irrelevant if there is no interference. We will use $v^{\boldsymbol{\pi}}(\boldsymbol{s})$ to denote the sum of the value functions of all BSs under the frequency reuse policy $\boldsymbol{\pi}$.

Definition 1. *In a stochastic game a Nash equilibrium (NE) is a set of policies $\boldsymbol{\pi}^* = (\pi^{*1}, \ldots, \pi^{*b}, \ldots, \pi^{*B})$ such that for all $\boldsymbol{s} \in \mathbf{S}$, $\forall \pi^b \in \mathcal{P}^b$ ($\mathcal{P}^b$ is the set of all feasible frequency reuse policies for BS b) and $b = 1, \ldots, B$,*

$$v^{\boldsymbol{\pi}^*}(\boldsymbol{s}) \geq v^{\boldsymbol{\pi}'}(\boldsymbol{s}),\ \textit{where}\ \boldsymbol{\pi}' = (\pi^{*1}, \ldots, \pi^{b}, \ldots, \pi^{*B}). \tag{12}$$

We will next establish the existence of and convergence to a Nash equilibrium for our non cooperative Markov game among the BSs.

A. Existence of and convergence to a Nash equilibrium for our proposed non cooperative Markov game

Theorem 1. *Consider a non cooperative Markov game where each BS in the network is autonomously learning a frequency*

reuse policy to mitigate interference. There exists a Nash equilibrium, possibly not unique, for the game under all three reward modes (G1), (G2) and (G3), with each BS's agent converging to an NE frequency reuse policy.

Proof. We have a non cooperative Markov game where each BS agent determines its optimal policy in response to the either the sum reward as in (G1) or a proxy as in (G2), in (G3) that reflects the reward of all BSs in the network. This is an B-player general sum stochastic game known as the Nash Q learning algorithm [26]. The convergence of Nash Q learning has been established in [26, Sec 3.2], when the following three conditions are satisfied: (i) each action state is visited infinitely often, (ii) the learning rate step size satisfies $0 \leq \alpha_t < 1, \sum \alpha_t = \infty, \sum \alpha_t^2 < \infty$ and (iii) the game has either a global optimum or a saddle point. While the first two conditions can be easily satisfied by the choice of learning hyper parameters, the last condition follows from the fact that an n-player game with finite actions has at least one Nash equilibrium with mixed strategy [29]. Note that both (G2) and (G3) use the *relative* downlink rate in (10) as the training reward which indirectly reflects on how well other BSs in the network are doing. Specifically, a smaller *relative* downlink rate over a prolonged time duration implies that the neighboring BSs are causing more interference due to their users' queues being active. ■

IV. Main Results

We shall first introduce a few definitions needed to present our main theoretical results. Next we define a notion of the *capacity region* of our proposed Probabilistic Frequency Reuse (PFR) for the non cooperative Markov game. By *capacity region* we refer to the set of all user arrival rate vectors that the network is able to support with stable queues. Next we establish a capacity order among the three game settings (G1), (G2) and (G3) according to their learnt value functions.

A. Network stability under interference mitigation polices

We begin by defining a notion of capacity for the network given a frequency reuse policy $\boldsymbol{\pi} = (\pi^b, b \in \mathcal{B})$ which characterizes the set of possible long term downlink transmission rates which are achievable under two assumptions (a) all users' transmit queues are backlogged, and (b) all BSs make use of the all resources made available by their respective frequency reuse policies, and thus offer the worst case interference according to their frequency reuse policy.

Recall that $\boldsymbol{\pi}(\mathbf{s})$ denotes the probability that the frequency reuse policy across all the BSs is in state $\boldsymbol{s}$. Further, given the frequency reuse and channel states of the network $\mathbf{s}, \mathbf{g}$ and $s_k^b = 1$ (resource k is available at base station b), $\phi_{uk}^b(\boldsymbol{s}, \boldsymbol{g})$ denotes the fraction of time resource k is allocated to user $u \in \mathcal{U}^b$, and thus under the assumption (b) in the paragraph above, we have that $\sum_{u \in \mathcal{U}^b} \phi_{uk}^b(\boldsymbol{s}, \boldsymbol{g}) = 1$. This corresponds to a static splitting resource allocation policy across the network when the network is in state $\boldsymbol{s} \in \mathbf{S}, \boldsymbol{g} \in \mathbf{G}$. Let $\mathcal{F}$ denote the set of such feasible splittings for all possible network states,

$$\mathcal{F} = \left\{ \boldsymbol{\phi} : \forall b \in \mathcal{B}, \forall \boldsymbol{s}, \boldsymbol{g}, \text{if } s_k^b = 1 \text{ then } \sum_{u \in \mathcal{U}^b} \phi_{uk}^b(\boldsymbol{s}, \boldsymbol{g}) = 1 \right\}. \tag{13}$$

Suppose $\boldsymbol{\phi}$ is a feasible static splitting, then one could come up with a lower bound on the downlink rate under a frequency reuse $\boldsymbol{\pi}$ for each user $u \in \mathcal{U}^b$ given by,

$$\mu_u^{\infty}(\boldsymbol{\pi}, \boldsymbol{\phi}) = \mathbb{E}^{\boldsymbol{\pi}, \chi} \left[\sum_{k=1}^{N} \phi_{uk}^b(\mathbf{S}, \mathbf{G}) c_{uk}(\mathbf{S}, \mathbf{G}) \right]. \tag{14}$$

where $\boldsymbol{\pi}$ and χ correspond to the distributions of the network's frequency reuse and channel states $\mathbf{S}$ and $\mathbf{G}$ respectively. We further let $\mu^{\infty}(\boldsymbol{\pi}, \boldsymbol{\phi}) = (\mu_u^{\infty}(\boldsymbol{\pi}, \boldsymbol{\phi}), u \in \mathcal{U})$.

Definition 2. *Given a frequency reuse policy $\boldsymbol{\pi}$, we define the **saturated network capacity** region $\mathcal{C}_{\pi}^{\infty}$ as follows*

$$\mathcal{C}_{\boldsymbol{\pi}}^{\infty} \triangleq \{ \boldsymbol{r} : \mathbf{0} \preceq \boldsymbol{r} \preceq \boldsymbol{\mu}^{\infty}(\boldsymbol{\pi}, \boldsymbol{\phi}), \quad \boldsymbol{\phi} \in \mathcal{F} \}.$$

Notation: $\mathcal{C}_{\boldsymbol{\pi}}^{\infty, b}$ denotes the *saturated capacity* region for BS b, $(\cdot)^{\circ}$ denotes *interior* of a set and $\boldsymbol{\lambda}^b$ denotes the user arrival rate vector of the arrival rate for BS b. $\boldsymbol{\lambda}^{-b}$ denotes the arrival rate vector of users at all BSs in the network except BS b.

We further define the *capacity region* for saturated networks under all possible Markovian frequency reuse polices $\mathcal{P}$ as

$$\mathcal{C}^{\infty} = \bigcup_{\boldsymbol{\pi} \in \mathcal{P}} \mathcal{C}_{\boldsymbol{\pi}}^{\infty} \quad . \tag{15}$$

Remark. It is easy to show that $\mathcal{C}_{\boldsymbol{\pi}}^{\infty}$ is convex owing to the convexity of the set of all possible static splits $\mathcal{F}$. However, $\mathcal{C}^{\infty}$ need not be convex. Indeed one could design $\boldsymbol{\pi}_1$, $(\boldsymbol{\pi}_2)$ which allocate resources only to b_1, (b_2), respectively. In this case it is not feasible to achieve a convex combination of $\mathcal{C}_{\boldsymbol{\pi}_1}$ and $\mathcal{C}_{\boldsymbol{\pi}_2}$ as that calls for a scheduler aware frequency reuse policy which goes against our separation of concerns paradigm.

Lemma 1. *Consider a network where the users' queues across the BSs have iid arrivals with mean $\boldsymbol{\lambda}$ such that there exists a π and $\boldsymbol{\phi}$ such that $\boldsymbol{\lambda} < \boldsymbol{\mu}^{\infty}(\boldsymbol{\pi}, \boldsymbol{\phi}) \in \mathcal{C}^{\infty}$, then the network is stable under the reuse policy $\boldsymbol{\pi}$ with static splitting rule $\boldsymbol{\phi}$.*

Proof. For a frequency reuse policy $\boldsymbol{\pi}$ and static split rule $\boldsymbol{\phi}$ consider the standard Lyapunov function $V(\mathbf{Q}) = \sum_{b \in \mathcal{B}} (\boldsymbol{q}^b)^T \boldsymbol{q}^b$. Note that the Lyapunov drift of this network is at least as the large as that of the case where all user queues in the network are infinitely backlogged. One can easily check for Foster's stability criterion (in the infinitely backlogged users case) to show that $V(\mathbf{Q})$ has a negative drift when $\exists \boldsymbol{\mu}(\boldsymbol{\pi}, \boldsymbol{\phi}) \in \mathcal{C}^{\infty}$ such that $\boldsymbol{\lambda} < \boldsymbol{\mu}(\boldsymbol{\pi}, \boldsymbol{\phi})$. ■

Lemma 2. *For a given frequency reuse policy $\boldsymbol{\pi}$, if the arrival rate $\boldsymbol{\lambda}^b$ at BS b is such that $\boldsymbol{\lambda}^b \in (\mathcal{C}_{\boldsymbol{\pi}}^{\infty, b})^{\circ}$, then assuming base station b employs max weight scheduling while all other BSs are saturated, the system is stable.*

Proof. Max weight scheduling algorithm [30] is throughput, i.e., stabilizes the user queue lengths, whenever feasible. Stability of max weight scheduler for a single base station has been established for iid arrivals and Markovian channel variations in [31]. The channel variations as seen at BS b in our setting are iid across all BSs in the network and the frequency reuse policy $\boldsymbol{\pi}$ which determines the availability of resources at each BS is Markovian. Therefore, the channel variations seen at BS b are Markovian. Furthermore arrivals at user queues are assumed independent throughout the network, allowing us to invoke result [31], to show stability. ■

Intuitively from a BS's perspective, saturation of neighboring BSs' users' queues corresponds to a worst case in terms of interference and thus the capacity that can be achieved for its users. One would expect if one relaxes the saturation assumption that one would still achieve stability. The following result shows that this is the case for a network where the arrivals at all BS/users lie in the interior of $\mathcal{C}^{\infty}$.

Theorem 2. *For a given frequency reuse policy $\boldsymbol{\pi}$, if the arrivals to the user queues in the network satisfy $\boldsymbol{\lambda} \in (\mathcal{C}_{\boldsymbol{\pi}}^{\infty})^{\circ}$, then max weight schedulers at each BS will stabilize it's users' queues.*

Proof. Suppose $\boldsymbol{\lambda} \in (\mathcal{C}_{\boldsymbol{\pi}}^{\infty})^{\circ}$, by Lemma 2 we know that each base station operating under max weight with the neighbors saturated would be stable, as shown by defining a quadratic Lyapunov function [32, Section 5] and showing it has negative drift outside a finite set of queue states. We note that if neighboring BSs are not saturated the drift at the base station would only be larger, because this would reduce the interference seen at the base station. To prove stability of the overall network we can consider a sum of the Lyapunov functions across BSs, since the state of the network as a whole is Markovian. Note that the sum of the Lyapunov functions has a larger service rate and thus a larger negative drift as compared to that of the same network with infinitely backlogged users. Thus it should be clear that user queues at each BS are stable even when the neighboring BSs are not saturated. ■

Remark. Local reward based (as in (G2), (G3)) distributed learning of frequency reuse policy $\boldsymbol{\pi}$ is *effective* only when the metric represents the relative performance of each user with reference to the no interference scenario.

Theorem 3. *For a given scheduling policy $\boldsymbol{h}$, let $\boldsymbol{\pi}_g^*$, $\boldsymbol{\pi}_l^*$ denote possibly not unique frequency reuse policies which are Nash Equilibria learnt using the global (G1) and local (G2) reward games, respectively. For every local frequency reuse policy $\boldsymbol{\pi}_l^*$ there exists a global frequency reuse policy $\boldsymbol{\pi}_g^*$ such that $v^{\boldsymbol{\pi}_g^*}(\boldsymbol{s}) \geq v^{\boldsymbol{\pi}_l^*}(\boldsymbol{s})$.*

Proof. Suppose we can pick a specific local Nash Equilibrium frequency reuse policy $\boldsymbol{\pi}_l$ for our network, let us choose a particular BS in the network. This selected BS is then allowed to learn a frequency reuse policy based on the global reward of the network. Clearly, this can only improve the rewards of the BS and hence the overall network, since the BS agent is now able to learn a policy based on rewards for it's actions using the overall network reward, i.e., $v^{\boldsymbol{\pi}_g^*}(\boldsymbol{s}) \geq v^{\boldsymbol{\pi}_l^*}(\boldsymbol{s})$. Therefore, for every $\boldsymbol{\pi}_l^*$ one can construct a global policy $\boldsymbol{\pi}_g^*$ that achieves better rewards. ■

V. Simulations

A simple network as in Fig. 3 with four BSs is considered. Each BS serves 5 users and has access to 5 resources (3.5-3.55 GHz with RB bandwidth 10 MHz). Users are dropped uniformly at random over each square region and associated to the nearest BS. For all three game settings (G1), (G2) and (G3) a DQN is employed at each BS to learn its frequency reuse policy π^b based on the rewards generated. The DQN learns a Q table [33] which keeps track of the mean discounted reward as a function of the (*state*, *action*) pair of each BS. The *state* of every BS agent is given by the binary vector corresponding to the frequency reuse state. The *action* corresponds to the resources selected by each BS for transmission for the next time slot, with 1(0) to denote whether a resource can be (cannot be) assigned to a user. During the training phase each DQN performs exploration and exploitation to learn a policy π^b that maximizes the long term discounted rewards. The Keras Adam optimizer [34] was used to implement the DQN with the following hyperparameters: exploration rate $\epsilon \in [1, 0.01]$, exploration decay factor 0.995, a learning rate of 0.001 and a reward discount factor $\gamma = 0.95$. Algorithm 1 gives the DQN training logic used at each of the BSs. Open AI gym [35] was used to simulate the wireless network environment for multi-agent RL.

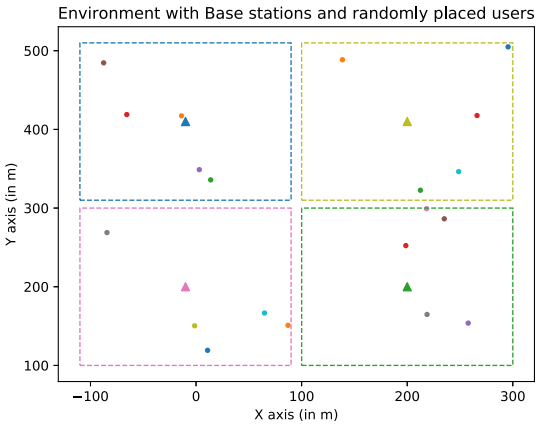

Fig. 3: The network configuration used for simulations with four BSs (triangles), each associated with 5 users (dots).

The SINR of a user scheduled to transmit, is calculated using (2). The transmit power of the BS (user) is 2 W (100 mW) with noise power set to -104 dB. $\hat{c}_u$ is the estimated data rate for user u per RB, based on channel quality (fading and path loss). We use bounded standard path loss $\max[1, r^{-\alpha}]$ with exponent $\alpha = 3$ to model channel gains. Note that we use $|h_i|^2 = 1$ for the simulations, since we assume that the Channel State Information (CSI) is available at the

transmitter. This is a reasonable assumption given that the BSs can evaluate the CSI for each of its associated users and hence the channel fading $|h|^2$ is not an unknown quantity under flat fading with CSI.

Algorithm 1: Training DQN of each agent

```
initialize policy π^b;
while training do
    generate random arrivals;
    schedule resources using policy π^b;
    save training data (state, action, reward, next state) to
     batch;
    if batch memory full, train DQN using batch data ;
    update policy π^b, choose action with max Q-value ;
    if iterations > maxIterations then
        done = True;
    end
end
```

At each BS the DQN provides a list of available resources to the scheduler that can be allocated to users. Each BS uses a Max weight scheduler (predetermined scheduler $\boldsymbol{h}$) to determine the users to be assigned available RBs. The weight for each user is calculated based on both the current user queue size N_u and the estimated downlink rate $\hat{c}_u$ as, $w_u = N_u\hat{c}_u$. A BS assigns RBs iteratively to its users as follows. The user with maximum weight is assigned the best channel available and then user weights are reevaluated based on updated queue size accounting for potential packet transmissions. Specifically, suppose user u was assigned a channel with rate $\hat{c}_u$ for transmission, then the user weight w_u is calculated with an updated queue size $N_u = N_u - f(\hat{c}_u)$, where $f(\cdot)$ is a non decreasing function that denotes the number of packets transmitted as a function of the downlink rate $\hat{c}_u$. We use a piece wise linear function $f(x) = \lfloor \log_2(1+x) \rfloor$ to determine the number of packets transmitted, as a function of the rate, $\lfloor \log_2(1+\text{SINR}) \rfloor$, where $\lfloor \cdot \rfloor$ is the floor operator. The minimum SINR threshold below which no packets can be reliably transmitted is set to 0dB.

The training algorithm for the DQN agent at each BS is shown in Algorithm 1. We use batch training for the DQNs, hence, a single iteration in the DQN training is equivalent to multiple time steps in real time resource scheduling. We use max-weight scheduling with frequency reuse 1 as a bench mark to gauge the performance of our multi agent RL framework. In order to find the "best" that one could do with a greedy strategy, we also include plots for an *oracle-aided* centralized benchmark where each BS completely knows the interference that will be seen as resources are allocated in the network. For the *centralized* benchmark, we use full frequency reuse at each BS in the network and a sequential scheduling order, wherein the n^{th} BS uses max weight scheduling of resources but is completely aware of the exact interference caused by the scheduling of the previous $n-1$ BSs.

Remark: We observed that the frequency reuse states learned across BSs based on $\boldsymbol{\pi}_g$ are more strongly correlated than those for $\boldsymbol{\pi}_l$. This can be attributed to the fact that the global reward reflects complete information about the network performance as compared to local rewards. Consequently, resource selection patterns learnt using the global reward (G1) better mitigate interference as compared to (G2) or (G3).

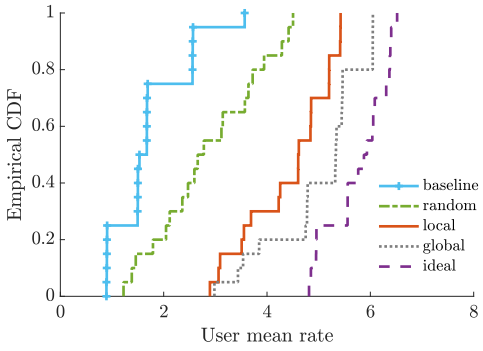

Fig. 4: User mean rate CDF for all policies.

The empirical CDF of the users' mean rate is shown in Fig. 4. It can be seen that even the simplest and most practical random action game (G3) results in better mean rate for users. Specifically, the global reward, local reward and random action games show a 4.8%, 14.8% and 25% improvement in the total rate delivered to all users, when compared to the baseline.

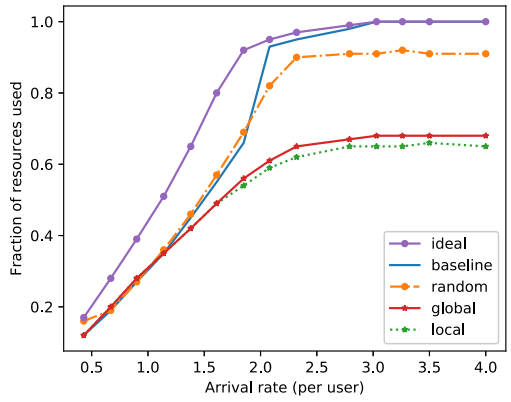

Fig. 5: Fraction of resources used on average across the network, as a function of the arrival rate per user

Fig. 5 shows the mean fraction of resources selected for downlink transmission in the network as a function of the arrival rate at each of the users' queues. Observe that when compared to the random action game, the global and local reward based policies learn to use a better frequency reuse strategies, which helps them achieve better throughput Fig. 4. Also, note that in all three settings considered, initially the fraction of resources allocated to users increases with an increase in the user packet arrival rate. However, beyond a critical level, the fraction of resources used almost saturates to a constant which is not 100%. While the global (G1)

and local reward (G2) games demonstrate a 32% and 34% improvement in energy efficiency through better resource utilization (resource positioning), the random action game (G3) still shows a 9% improvement in energy efficiency.

A further *interesting* result from experimenting with our proposed architecture is as follows. When all the BSs in the network except one, say BS 1, were configured to act greedily, that is employ frequency reuse 1, then using the proxy metric in (10) to train the DQN of BS 1 results in the agent to learn **not** to backoff but simply use all its RBs (frequency reuse 1). This behaviour demonstrates that our proposed algorithm is capable of learning the right policy in an adversarial setup. Another experiment to test the proposed architecture involved moving the BSs further away from each other (including their user locations), and it was observed that the DQN agents at each of the BS learn to use frequency reuse 1 as expected. Additional simulation results were not included due to lack of space.

VI. Conclusion

We have proposed two key concepts: First a separation of concerns where one fixes the base station scheduler and optimizes a frequency reuse policy for the given scheduler. Second, the use of a proxy reward metric that accounts for the interference coupling among base stations during the learning process. Also, the learning algorithm requires no information regarding the network topology, interference graph or user traffic dynamics. Furthermore, the training duration can be substantially reduced by a simplified action space in the *random action game*. Interesting future research directions include understanding the system behavior in the presence of user mobility and being able to manage delay sensitive traffic with finite queue size.

Acknowledgements

The authors would like to thank Harish Vishwanathan from Nokia Bell Labs and Chris Dick from NVIDIA, affiliates of the 6G@UT center within the Wireless Networking and Communications Group at The University of Texas at Austin, for all the initial discussions that helped formulate this problem. This work was also partially supported by NSF Grant CNS-1910112.

References

[1] A. Lozano, R. W. Heath, and J. G. Andrews, "Fundamental Limits of Cooperation," *IEEE Trans. on Info. Theory*, vol. 59, no. 9, 2013.

[2] N. Naderializadeh, et al., "Resource management in wireless networks via multi-agent deep reinforcement learning," 2020.

[3] M. Bennis and D. Niyato, "A Q-learning Based Approach to Interference Avoidance in Self-Organized Femtocell Networks," in *2010 IEEE Globecom Workshops*, pp. 706–710, Dec 2010.

[4] N. Zhao, et al., "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. on Wireless Comm.*, vol. 18, Nov 2019.

[5] X. Chen, et al., "Stochastic Power Adaptation with Multiagent Reinforcement Learning for Cognitive Wireless Mesh Networks," *IEEE Trans. on Mob. Comp.*, Nov 2013.

[6] H. Ye, et al., "Deep Reinforcement Learning Based Resource Allocation for V2V Comm.," *IEEE Trans. on Veh. Tech.*, vol. 68, April 2019.

[7] Y. S. Nasir and D. Guo, "Multi-Agent Deep Reinforcement Learning for Dynamic Power Allocation in Wireless Networks," *IEEE J. on Sel. Areas in Comm.*, vol. 37, no. 10, pp. 2239–2250, 2019.

[8] M. Simsek, M. Bennis, and İ. Güvenç, "Learning Based Frequency- and Time-Domain Inter-Cell Interference Coordination in HetNets," *IEEE Trans. on Veh. Tech.*, Oct 2015.

[9] J. Jang, et al., "Learning-Based Distributed Resource Allocation in Asynchronous Multicell Networks," in *International Conf. on Information and Comm. Tech. Convergence (ICTC)*, Oct 2018.

[10] Y. Wei, et al., "User Scheduling and Resource Allocation in HetNets With Hybrid Energy Supply: An Actor-Critic Reinforcement Learning Approach," *IEEE Trans. on Wireless Comm.*, Jan 2018.

[11] F. Shah-Mohammadi and A. Kwasinski, "Deep Reinforcement Learning Approach to QoE-Driven Resource Allocation for Spectrum Underlay in Cognitive Radio Networks," in *2018 IEEE International Conference on Comm. Workshops (ICC Workshops)*, May 2018.

[12] B. Rengarajan and G. de Veciana, "Architecture and Abstractions for Environment and Traffic-Aware System-Level Coordination of Wireless Networks," *IEEE Trans. on Net.*, 2011.

[13] T. K. Thuc, E. Hossain, and H. Tabassum, "Downlink Power Control in Two-Tier Cellular Networks With Energy-Harvesting Small Cells as Stochastic Games," *IEEE Trans. on Comm.*, 2015.

[14] F. Meshkati, et al., "Energy-efficient Resource Allocation in Wireless Networks with Quality-of-Service Constraints," *IEEE Trans. on Comm.*, vol. 57, no. 11, pp. 3406–3414, 2009.

[15] Z. Han, et al., "Noncooperative Power-Control Game and Throughput Game Over Wireless Networks," *IEEE Trans. on Comm.*, 2005.

[16] Q. Ni and C. C. Zarakovitis, "Nash Bargaining Game Theoretic Scheduling for Joint Channel and Power Allocation in Cognitive Radio Systems," *IEEE J. on Sel. Areas in Comm.*, 2012.

[17] S. Ramakrishnan and V. Ramaiyan, "Completely Uncoupled Algorithms for Network Utility Maximization," *IEEE Trans. on Net.*, 2019.

[18] Y. Ozcan and C. Rosenberg, "Uplink Scheduling in Multi-Cell OFDMA Networks: A Comprehensive Study," *IEEE Trans. on Mob. Comp.*, 2020.

[19] A. Asadi and V. Mancuso, "A Survey on Opportunistic Scheduling in Wireless Comm.," *IEEE Comm. Surveys Tutorial*, 2013.

[20] E. Evdokimova, et al, "Coupled Queues with Customer Impatience," *Performance Evaluation*, vol. 118, pp. 33–47, 2018.

[21] J. Pender, "An analysis of nonstationary coupled queues," *Telecommunication Systems*, vol. 61, no. 4, pp. 823–838, 2016.

[22] C. Knessl and J. A. Morrison, "Two Coupled Queues with Vastly Different Arrival Rates: Critical Loading Case," *Advances in operations research*, vol. 2011, pp. 1–26, 2011.

[23] S. C. Borst, et al., "The asymptotic workload behavior of two coupled queues," *Queueing systems*, no. 1-2, 2003.

[24] C. Knessl and J. A. Morrison, "Asymptotic Analysis of Two Coupled Large Capacity Queues with Vastly Different Arrival Rates," *Applied Mathematics Research Express*, 2012.

[25] S. Borst, et al., "Stability of Parallel Queueing Systems with Coupled Service Rates," *Discrete Event Dynamic Systems*, 2008.

[26] J. Hu and M. P. Wellman, "Nash Q-Learning for General-Sum Stochastic Games," *J. Mach. Learn. Res.*, vol. 4, p. 10391069, Dec. 2003.

[27] P. Kazemi, H. Al-Tous, C. Studer, and O. Tirkkonen, "Snr prediction in cellular systems based on channel charting," in *2020 IEEE Eighth International Conference on Communications and Networking (ComNet)*, pp. 1–8, 2020.

[28] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine Learning Proceedings*, 1994.

[29] J. E. Harrington Jr., "Essays on the foundations of game theory," *Managerial and Decision Economics*, vol. 12, no. 4, pp. 329–334, 1991.

[30] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *IEEE Trans. on Auto. Cntrl*, Dec 1992.

[31] A. L. Stolyar, "Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic," *Ann. Appl. Probab.*, vol. 14, pp. 1–53, 02 2004.

[32] M. Andrews, et al., "Scheduling in a queuing system with asynchronously varying service rates," vol. 18, p. 191217, Apr. 2004.

[33] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[34] "Keras Optimizers." https://keras.io/api/optimizers/.

[35] R. Lowe, et al., "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," *Neural Info. Proc. Systems (NIPS)*, 2017.