Automatica 169 (2024) 111817

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Distributed learning in congested environments with partial information☆

Amir Leshem [a,*], Vikram Krishnamurthy [b], Tomer Boyarski [a]

[a] Faculty of Engineering, Bar-Ilan University, Ramat-Gan 52900, Israel
[b] School of Electrical and Computer Engineering, Cornell University, Ithaca, 14853, USA

ARTICLE INFO

Article history:
Received 15 August 2022
Received in revised form 27 March 2024
Accepted 18 June 2024
Available online 8 August 2024

Keywords:
Distributed learning
Congestion games
Learning in dense environments
Learning in games
Poly-logarithmic regret

ABSTRACT

How can non-communicating agents learn to share congested resources efficiently? This is a challenging task when the agents can access the same resource simultaneously (in contrast to multi-agent multi-armed bandit problems) and the resource valuations differ among agents. We present a fully distributed algorithm for learning to share in congested environments and prove that the agents' regret with respect to the optimal allocation is poly-logarithmic in the time horizon. Performance in the non-asymptotic regime is illustrated in numerical simulations. The distributed algorithm has applications in cloud computing and spectrum sharing.

1. Introduction

Suppose N agents need to share M resources where $N \gg M$, i.e., in a congested environment. The utility of each agent n at each time instant t depends on the resource m_n^t it chooses and is inversely proportional to the number of other agents that choose the same resource at the same time, i.e., when each agent gets an equal share of the resource. The agents do not know the value of each resource. Each agent is aware of the resource-sharing structure but can only obtain a noisy realization of its utility corrupted by sub-Gaussian noise.[1] This problem is a generalization of the multi-armed multi-agent bandit problem for distributed resource sharing (Bistritz & Leshem, 2018b, 2020). The problem can be considered as learning of a congestion game with incomplete information. Like most games, resource-sharing among agents can benefit significantly from cooperation (Owen, 1968). This is especially important when there is no central management of the resources. Resource sharing is a challenging problem when communication between agents is limited or does not exist. A further complication arises when agents need to learn their individual resource valuations. An example of unknown valuations is the case of multiple servers having different hardware architectures, e.g., GPU-based machines and vector processors, a different number of cores, size of memory, and different communication links to the servers are some examples.

Resource sharing can be modeled as a non-cooperative game (Owen, 1968), specifically, a congestion game (Monderer & Shapley, 1996; Rosenthal, 1973). More specifically, resource sharing can be modeled as a congestion game with agent-specific utilities (Milchtaich, 1996) that arises when agents are scattered in the physical world. These games are used to model a wide variety of applications including routing, load-balancing, and spectrum allocation (Cheng et al., 2013; Pradelski & Young, 2012; Suri et al., 2004). Using game theory, it is possible to design adaptive agents whose actions optimize the overall welfare, where the welfare of a game is defined as the sum of the rewards, even when based on partial and imprecise information. Best response algorithms applied to congestion games converge to Pure Nash Equilibria (PNE), which may have low welfare. One way to deal with this problem is to bound the ratio between the best and worst PNE (Koutsoupias & Papadimitriou, 1999). A second way is to search for an efficient PNE (Pradelski & Young, 2012). A third way is to perform the welfare maximization of congestion games in a central unit (Blumrosen & Dobzinski, 2007). A fourth way is to simplify the discussion by not considering player-specific rewards (Cheng et al., 2013). Finally, the discussion can be restricted to a special class of congestion games called collision

☆ Amir Leshem was partially supported by ISF grant 2197/22 and the Bar-Ilan DSAI research center. Vikram Krishnamurthy was partially supported by ARO grant W911NF-24-1-083 and NSF grant CCF-2112457 and NSF grant CCF-2312198. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Rong Su under the direction of Editor Christos G. Cassandras.

* Corresponding author.
E-mail addresses: amir.leshem@biu.ac.il (A. Leshem), vikramk@cornell.edu (V. Krishnamurthy), tomer.boyarski@gmail.com (T. Boyarski).

[1] The sub-Gaussian family of distributions (see Appendix A.1) comprises many distributions including the Gaussian, any finitely supported distribution (e.g. uniform), Bernoulli, and many others (see Wainwright (2019)).

https://doi.org/10.1016/j.automatica.2024.111817

games, where all colliding players receive zero utility (Bistritz & Leshem, 2019). However, in collision games, when the number of resources is smaller than the number of agents, some agents will receive zero utility.

Typically, achieving the social optimum requires communication between agents. Surprisingly, it has been shown recently that for collision-type multi-agent multi-armed bandit problems, a distributed assignment problem can be solved without explicit information exchange between agents, as long as the number of resources is larger or equal to the number of agents. Examples include the Game of Thrones algorithm (Bistritz & Leshem, 2018b, 2020), exploiting the auction algorithm (Bertsekas, 1979) to obtain logarithmic regret in the multi-agent multi-arm bandit with message exchange between the agents (Kalathil et al., 2014), using the distributed auction algorithm (Naparstek & Leshem, 2013) which exploit properties of the wireless channel to obtain a distributed learning with no coordination (Zafaruddin et al., 2019), swap-based algorithms for stable configurations (Avner & Mannor, 2019; Darak & Hanawal, 2019), or to signal resource valuations (Boursier et al., 2019; Bubeck et al., 2019; Tibrewal et al., 2019) and the musical chairs algorithm (Rosenski et al., 2016). Similarly, algorithms for the adversarial case have been studied in Alatur et al. (2020). In these papers, it is assumed that agents who choose the same arm simultaneously receive no reward.

Motivated by applications in spectrum collaboration in ad-hoc wireless networks (Avner & Mannor, 2019; Bistritz & Leshem, 2018a; Bubeck et al., 2019; Darak & Hanawal, 2019; Tibrewal et al., 2019; Zafaruddin et al., 2019), cloud computing (Tang et al., 2018) and machine scheduling (Tang et al., 2018), we deal with the heavily congested regime together with utilities that depend non-linearly on the number of sharing agents. When multiple agents simultaneously choose the same arm they suffer a penalty. To overcome this, arm-sharing protocols can be devised, and the utility of each agent depends on the load on the arm. Recently, we have shown that a time division structure can be implemented using opportunistic carrier sensing combined with the auction algorithm (Boyarski et al., 2023). However, this requires a more complicated communication protocol. In Magesh and Veeravalli (2021) a collision-based solution is proposed under a slightly different setting.

The algorithms and analysis in this paper depart significantly from existing works in that we consider simultaneous resource sharing in heavily loaded systems with multiple non-communicating agents. Specifically, we develop a learning framework for incomplete information congestion games with non-linear utilities. To that end, we construct a novel algorithm (Estimate, Negotiate, Exploit), dividing the learning into epochs of increasing length. Each epoch has a constant exploration phase followed by a negotiation phase which is a poly-logarithmic fraction of the following exploitation phase. By proving that the probability of error in the negotiation phase decreases super-exponentially we obtain the main result: A poly-logarithmic regret in Theorem 4.1. That is, the regret R grows with time T as $R = O(\log_2^{3+\delta}(T))$. where $0 \leq \delta \leq 1$.

Our proposed algorithm extends the work of Marden et al. (2014) to incomplete information games and Bistritz and Leshem (2018a) to the heavily congested case where there are fewer resources than agents and there is no collision information but only rewards which depend on the load since multiple agents access each resource simultaneously. Another novel feature of our algorithm is the identification of specific states, which allows us to accelerate the algorithm, reduce the effective state space used in the negotiation phase and therefore allow us to apply the algorithm for a larger number of agents.

The rest of this paper is organized as follows: Section 2 sets up the model formulation and describes the problem. Section 3 describes our novel learning algorithm. Section 4 performs a regret analysis of the learning algorithm. In Section 5 we discuss the complexity of the algorithm, and the structure of the state space. Following this, we propose a significant reduction in the computational cost as well as the convergence time, by devising a novel step for distributed identification of the all-content state without explicit communication between the agents. This in turn allows us to consider the conditional stationary distribution of the perturbed Markov chain conditioned on all agents being in the content state. In Section 6 we give numerical examples to illustrate the regret of the proposed algorithm compared to the distributed Upper Confidence Bound algorithm and random allocations. Details of the proofs are provided in the Appendix.

2. Distributed cooperative sharing of congested resources

In this section, we define the resource sharing problem. We assume that each resource is equally shared among the agents who choose it, e.g., via a round-robin mechanism. This is the simplest mechanism for sharing the resource when the resource is required continuously by all agents. Suppose N agents are sharing M resources where $N \gg M$. This makes the resource sharing much more challenging than the case $N \leq M$; Yet this model is important in spectrum sharing and cloud computing applications as discussed in Section 1.

We assume that time is slotted with $t = 1, 2, \ldots, T$ indexing the time slots (discrete time) and agents are synchronized to the slots. The number of time slots T is unknown to the agents. The single resource chosen by agent n at time t is denoted m_n^t. The *allocation* at time t is

$$\mathbf{m}^t = (m_1^t, \ldots, m_N^t)$$

The *load* experienced by agent n at time t under allocation $\mathbf{m}^t$ is the number of agents who chose the same resource (including itself), i.e.,

$$\ell_n^t = \ell_n(\mathbf{m}^t) \triangleq \sum_{k=1}^{N} \mathbb{1}(m_k^t = m_n^t), \tag{1}$$

where m_k^t is the action of agent k at time t. The utility of agent n with resource m and load 1 is denoted by $U_{n,m,1}$. We assume that these utilities are nonnegative, bounded by $U_{\max}$, and do not evolve with time. The utility of agent n at time t, given the resource allocation vector $\mathbf{m}^t$ is

$$U_n(\mathbf{m}^t) = \frac{U_{n,m_n^t,1}}{\ell_n^t}. \tag{2}$$

The welfare W at time t with allocation $\mathbf{m}^t$ is the sum of the utilities over the N agents:

$$W^t = W(\mathbf{m}^t) \triangleq \sum_{n=1}^{N} U_n(\mathbf{m}^t) = \sum_{n=1}^{N} U_n^t. \tag{3}$$

The best and second-best welfares are denoted by

$$W^* \triangleq \max_{\mathbf{m}} W(\mathbf{m}) \qquad W^{**} \triangleq \max_{\mathbf{m} \neq \mathbf{m}^*} W(\mathbf{m}) \tag{4}$$

The sub-optimality gap is defined as:

$$\rho \triangleq \frac{W^* - W^{**}}{2N} \tag{5}$$

The optimal allocation is

$$\mathbf{m}^* \triangleq \arg\max_{\mathbf{m}} W(\mathbf{m}). \tag{6}$$

A crucial property of our model is that of *incomplete information*: agents are aware of the time-sharing structure of the utility

function that is inversely proportional to their load, but they cannot directly observe their utilities, and they do not know their unique utility function parameters $U_{n,m,1}$ $\forall 1 \leq m \leq M$. Instead, they observe noisy versions of their utilities known as *sample rewards*. The sample reward of agent n at time t with in allocation $\mathbf{m}^t$ is

$$r_n(\mathbf{m}^t) = U_n(m_n^t) + \nu_n^t, \tag{7}$$

where ν_n^t is zero-mean sub-Gaussian noise, i.i.d. in time and among agents with variance proxy b.[2]

Since the utilities of the agents with each resource and load 1 are independently distributed continuous random variables, drawn once at the beginning of the sharing process, the optimal allocation $\mathbf{m}^*$ is unique with probability 1.

The main performance metric of the above resource-sharing processes is the *regret*:

$$R \triangleq TW^* - \mathbb{E}\left(\sum_{t=1}^{T} W^t\right), \tag{8}$$

where the expectation $\mathbb{E}$ is taken with respect to the randomness in the rewards as well as the agents' choices.

3. Learning the optimal allocation

How can the optimal allocation $\mathbf{m}^*$ defined in (6) be learned by the agents in a distributed way? This section presents the *Estimation, Negotiation, and Exploitation (ENE)* learning algorithm that achieves a regret that is poly-logarithmic in the number of time steps

$$R = O(\log_2^{3+\delta}(T))$$

where $0 \leq \delta \leq 1$. The algorithm divides the T time slots of the sharing problem into J epochs of dynamic length. As in other related work, e.g., Bistritz and Leshem (2018a), Boursier et al. (2019), Tibrewal et al. (2019) and Zafaruddin et al. (2019), each epoch is further divided into phases whose length is also dynamic.

Our proposed algorithm has three phases: Estimation, Negotiation, and Exploitation.

(1) In the first phase each agent individually and distributedly *estimates* its utilities with any resource and any load.
(2) In the second phase agents *negotiate* over resources without direct communication.
(3) In the third phase agents *exploit* the allocation they distributedly decided upon in the previous phase. If the first and second phases are successful, the third phase is regret-free.

The algorithm is illustrated in Fig. 1. Each of the three phases is further divided into blocks. Intuitively, the purpose of the blocks is to average out the sub-Gaussian noise. There is an underlying clock that generates the time steps of the sharing process. We assume that the agents are synchronized to this clock. This assumption is typical in other papers in multi-agent distributed learning. The index n refers to an agent, m to a resource, and ℓ to the load. The reward during epoch j, phase i, block k, and time-step τ is denoted $r_{n,m,\ell}^{j,i,k,\tau} = U_{n,m,\ell} + \nu_n^{j,2,k,\tau}$.

Estimation phase: This phase has $M+1$ blocks. Each block has j time-steps.[3] In block k where $1 \leq k \leq M$ agent n accesses resource k with probability 1. As the number of epochs $j \to \infty$, the number of samples used for estimating the utilities also grows to infinity, and therefore, by the strong law of large numbers a consistent estimate of the utilities is possible, as shown below.

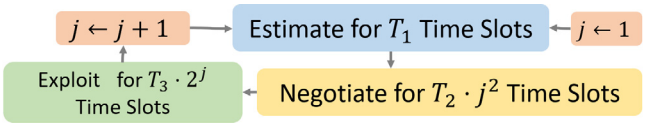

Fig. 1. The ENE algorithm structure. Each epoch j consists of three phases: Estimation, Negotiation and Exploitation, on time intervals $T_1(j), T_2(j), T_3(j)$, respectively.

Agent n then estimates its utility with resource k and load N as the average of its rewards from this block from all epochs until now:

$$\hat{U}_{n,k,N}^j = \bar{r}_n^{j,1,k} = \frac{\sum_{i=1}^{j}\sum_{\tau=1}^{i} r_n^{i,1,k,\tau}}{\frac{1}{2}j(j+1)} \xrightarrow{j\to\infty} U_{n,k,N} \tag{9}$$

In each time step of block $M+1$ each agent accesses the first resource with probability 1/2. Agent n denotes the average of its rewards from this block from all epochs until now by[4]

$$\bar{r}_n^{j,1,M+1} = \frac{\sum_{i=1}^{j}\sum_{\tau=1}^{i}\left(\mathbb{1}\left(m_n^{j,1,M+1,\tau}=1\right)\cdot r_n^{i,1,M+1,\tau}\right)}{\sum_{i=1}^{j}\sum_{\tau=1}^{i}\mathbb{1}\left(m_n^{j,1,M+1,\tau}=1\right)} \xrightarrow{j\to\infty} U_{n,1,N}\cdot 2\left(1-\frac{1}{2^N}\right) \tag{10}$$

Agent n estimates N in epoch j to be

$$\hat{N}_n^j = \frac{1}{\ln(1/2)}\ln\left(1-\frac{\bar{r}_n^{j,1,M+1}}{2\bar{r}_n^{j,1,1}}\right) \xrightarrow{j\to\infty} N \tag{11}$$

Recall that $\hat{N}_n^j$ denotes agent n's estimate of the number of agents and that $\hat{U}_{n,m,N}^j$ is its estimate of its own utility with resource m and maximal load. Agent n can now use these two quantities to compute its utility with any load ℓ:

$$\hat{U}_{n,m,\ell}^j = \frac{1}{\ell}\hat{N}\hat{U}_{n,m,N}^j \xrightarrow{j\to\infty} U_{n,m,\ell} \tag{12}$$

The allocation that maximizes the estimated utilities of epoch j is denoted by

$$\mathbf{m}^{*j} \triangleq \arg\max_{\mathbf{m}} \sum_{n=1}^{N} \hat{U}_{n,m_n,\ell_n(\mathbf{m})}^j \tag{13}$$

The estimated optimal utilities are

$$\begin{aligned} \hat{U}_n^{*j} &\triangleq \hat{U}^j_{n,m_n^{*j},\ell_n(\mathbf{m}^{*j})} \\ \hat{\mathbf{U}}^{*j} &\triangleq \left(\hat{U}_1^{*j},\ldots,\hat{U}_N^{*j}\right) \end{aligned} \tag{14}$$

Negotiation Phase: In the heart of the ENE algorithm is the Negotiation Phase, inspired by Marden et al. (2014). The Negotiation Phase of epoch j is divided into $j^{1+\delta/3}$ load-estimation-blocks[5] that are each composed of $j^{1+\delta/3}$ time-steps. Agent n in block k has a mood that is either Content or Discontent and denoted by $S_n^{j,2,k} \in \{C, D\}$. A Content agent is stable while a Discontent agent

[2] Some of the basic properties of sub-Gaussian random variables used in this paper are mentioned in the Appendix.

[3] In Magesh and Veeravalli (2021) the authors assume that the number of players is bounded and that whenever more than N^* access a resource, the utility is 0. Since we do not make this assumption, our concentration result requires, the number of samples for estimation to grow super-linearly.

[4] See Appendix.

[5] We believe that using the significantly more complicated techniques from Bistritz and Leshem (2020) we can use only $j^{\delta/3}$ Negotiation Blocks, thereby reducing to total regret of the algorithm to $O(\log^{2+\delta} T)$.

is unstable. The probability of an individual agent to enter such a Content and Stable state increases with its estimated utility. Therefore, the probability of the community to enter an all-content-all-stable state increases with the Welfare. On the other hand, the probability to exit an all-Content-all-stable state is constant and independent of the Welfare. Hence, during the tail of the Negotiation Phase, the community will spend most of its time in an optimal all-Content-all-stable state with high probability. Agents can then count which resources they visit most frequently during the tail of the Negotiation Phase and use these during the Exploitation Phase. The meta-parameter controlling the dynamics of the Markov chain is denoted by $\varepsilon \in (0,1)$. In the first block of the Negotiation Phase, all agents are Discontent. In block k, agent n performs the following actions distributedly and individually without direct communication with its peers:

(1) Choose a resource. A discontent agent n chooses a resource $m_n^{j,2,k}$ at epoch j, phase 2 (Negotiation) and block k uniformly at random:

$$\mathbb{P}\left(m_n^{j,2,k}=a\right)=\frac{1}{M},\ \forall\ 1\le a\le A. \tag{15}$$

A content agent chooses the same resource with high probability and will otherwise explore uniformly:

$$\mathbb{P}\left(m_n^{j,2,k}=a\right)=\begin{cases}1-\varepsilon^c & a=m_n^{j,2,k-1}\\ \frac{\varepsilon^c}{A-1} & a\neq m_n^{j,2,k-1}\end{cases} \tag{16}$$

where $c > N$ is a parameter of Algorithm 1. c ensures that an agent stays content for sufficiently long time so that other agents can become content as well (See the discussion in Section 5 for further insight.)

(2) Stay with this resource for the rest of this block and collect $j^{1+\delta/3}$ i.i.d. reward samples.

(3) Averages these reward and denotes the average by $\bar{r}_n^{j,2,k}$.

(4) Estimate load based on the utility estimation from the previous phase:

$$\hat{\ell}_n^{j,2,k}\leftarrow\arg\min_{1\le\ell\le N}\left|\bar{r}_n^{j,2,k}-\hat{U}^j_{n,m_n^{j,2,k},\ell}\right| \tag{17}$$

(5) Estimate utility based on the load estimation $\hat{\ell}_n^{j,2,k}$ and the utility estimation of the previous phase:

$$\hat{U}_n^{j,2,k}\leftarrow\hat{U}^j_{n,m_n^{j,2,k},\hat{\ell}_n^{j,2,k}} \tag{18}$$

(6) Choose a new Mood. If an agent was previously Content $S_n^{j,2,k-1}=C$, and its action and estimated utility have not changed $m_n^{j,2,k}=m_n^{j,2,k-1}\wedge\hat{U}_n^{j,2,k}=\hat{U}_n^{j,2,k-1}$, then it will remain Content with probability 1:

$$C\to C\ \text{w.p.1.} \tag{19}$$

If an agent was previously Discontent $S_n^{j,2,k-1}=D$, or changed its resource or estimated utility from the previous block $m_n^{j,2,k}\neq m_n^{j,2,k-1}\vee\hat{U}_n^{j,2,k}\neq\hat{U}_n^{j,2,k-1}$, its new Mood is chosen according to the following probability:

$$[C/D]\to\begin{cases}C & \text{w.p. } \varepsilon^{U_{\max}-\hat{U}_n^{j,2,k}}\\ D & \text{w.p. } 1-\varepsilon^{U_{\max}-\hat{U}_n^{j,2,k}}.\end{cases} \tag{20}$$

Exploitation Phase: The third and final phase of the ENE algorithm has only one block and 2^j time-steps. Each agent chooses individually and distributedly the resource it visited most frequently during the tail of the last Negotiation Phase:

$$m_n^{j,3}=\arg\max_a\sum_{k=(1-\alpha)j^{1+\delta/3}}^{j^{1+\delta/3}}\mathbb{1}(m_n^{j,2,k}=a) \tag{21}$$

where $0 < \alpha < 1$ is a forgetting factor. Choosing $\alpha = 0.5$ is sufficient. The agent then stays with this resource throughout the block, gathering rewards. If the first two phases were successful, this phase will be regret free. The complete ENE method is described in Algorithm 1.

Algorithm 1 The Estimation, Negotiation, and Exploitation algorithm at the individual agent level, to be performed fully distributedly and without communication between agents

Input: $\varepsilon > 0, \alpha \in (0,1), \delta > 0, c \ge N, c_1, c_2, c_3, c_4$ the initial lengths of the phases.
$Mood_n \leftarrow D$.
for $j = 1$ **to** J epochs **do**
  Payoff Estimation Phase
  for $m = 1$ **to** M **do**
    for $\tau = 1$ **to** $c_1 j$ **do**
      $m_n^{j,1,m,\tau} \leftarrow m$
    end for
    Estimate $U^j_{n,m,N}$ according to (9).
  end for
  for τ **to** j **do**
    $m_n^{j,1,M+1,\tau} = \begin{cases}1 & \text{w.p. } 1/2\\ \emptyset & \text{w.p. } 1/2\end{cases}$
  end for
  Calculate $\bar{r}_n^{j,1,M+1}$ according to (10).
  Estimate N according to (11).
  Estimate $U^j_{n,m,\ell}\ \forall 1\le m\le M, 1\le\ell\le N$ according to (12).
  Negotiation Phase
  $S_n^{j,2,0} \leftarrow D$
  for $k = 1$ **to** $c_2 j^{1+\delta/3}$ **do**
    Choose new resource $m_n^{j,2,k}$ according to (15) or (16).
    for $\tau = 1$ **to** $c_4 j^{1+\delta/3}$ **do**
      $m_n^{j,2,k,\tau} \leftarrow m_n^{j,2,k}$
    end for
    Calculate $\bar{r}_n^{j,2,k} \leftarrow \frac{1}{j^{1+\delta/3}}\sum_{\tau=1}^{j^{1+\delta/3}} r_n^{j,2,k,\tau}$.
    Estimate load $\hat{\ell}_n^{j,2,k}$ according to (17).
    Estimate utility $\hat{U}_n^{j,2,k}$ according to (18).
    Choose new Mood according to (19) or (20).
  end for
  Exploitation Phase
  Choose resource $m_n^{j,3}$ according to (21).
  for τ **to** $c_3 2^j$ **do**
    $m_n^{j,3,1,\tau} \leftarrow m_n^{j,3}$
  end for
end for

4. Regret analysis of ENE algorithm

In this section, we analyze the expected regret of the ENE Algorithm 1. We present the main Theorem, whose proof follows via a sequence of Lemmas (in the Appendix) to bound the probability of error for each error event.

Theorem 4.1. *For the resource sharing problem specified in Section 2 there exists a parameter*[6] $\varepsilon > 0$ *in algorithm 1 such that the regret of the ENE algorithm is upper-bounded by* $O\left(\log_2^{3+\delta}(T)\right)$.

Before providing the rigorous proof, we provide an outline that will assist the reader. We consider the first two phases as resulting in full regret, i.e., each agent suffers regret of $U_{\max}$.

[6] The choice of ε is related to the perturbed Markov chain used in the Negotiation Phase. Practically, we found that values between 10^{-3} and 0.1 perform satisfactorily.

Hence the contribution of these phases to epoch j's expected regret is $O\left(j^{2+\delta}\right)$ which is the length of these phases in epoch j. Summing up to epoch J this yields $O\left(J^{3+\delta}\right)$. Since exploitation grows exponentially, up to time T there are $O(\log T)$ epochs and the total regret from exploration and negotiation is $O\left(\log^{3+\delta} T\right)$ (This is proved in Lemma 4.2.) The main part of the proof is showing that the total regret from exploitation is bounded. This comprises two parts: Prove that the probability that the agents use sub-optimal actions at the end of the negotiation phase of epoch j decays exponentially at a rate faster than 2^{-j}. This is based on the analysis of the Markov chain and the proof is given in Appendix A.3. All that remains is to sum up the geometric series bounding the overall exploitation regret.

Proof. Let R_1, R_2 and R_3 denote the accumulated regret from the Estimation, $O\left(J^{3+\delta}\right)$ Negotiation, and Exploitation phases of all the epochs of the algorithm, respectively, such that $R = R_1 + R_2 + R_3$. Recall that Algorithm 1 operates over J epochs. By Lemma 4.3 $R_3 = O(J)$ and $R_1 + R_2$ is upper bounded by

$$NU_{\max} \sum_{j=1}^{J} \left(j(M+1) + j^{2+2\delta/3}\right) = O\left(J^{3+\delta}\right) \tag{22}$$

Furthermore, according to Lemma 4.2 $J \leq \log_2(T)$. □

Having sub-linear regret means that the ratio between the amount of time spent on sub-optimal allocations and the amount of time spent on the optimal allocation approaches zero as $T \to \infty$. Because the sharing process described in Section 2 is a variation on a single-agent multi-armed bandit problem, the optimal regret for this problem is $O(\log_2(T))$ according to Lai and Robbins (1985) and not far from ours.

Lemma 4.2. *The number of epochs J that Algorithm 1 operates satisfies $E < \log_2(T)$*

Proof. Ignoring the last epoch and the durations of the Estimation and Negotiation Phases produces $T \geq \sum_{j=1}^{J-1} 2^j = (2^J - 2)$.

Lemma 4.3. $R_3 = O(J)$.

Proof. The exploitation phase of epoch j will accumulate regret only if the following error event occurred:

$$E^{j,3} : \mathbf{m}^{j,3} \neq \mathbf{m}^* \tag{23}$$

That regret is upper bounded by $NU_{\max} 2^j$. Therefore:

$$R_{j,3} \leq NU_{\max} 2^j \mathbb{P}\left(E^{j,3}\right) \tag{24}$$

According to Lemma A.5 in the Appendix, the probability $\mathbb{P}\left(E^{j,3}\right)$ is $O(\exp(-j^{1+\delta/4}))$. Hence, $R_{j,3} = O(1)$. Finally, $R_3 = \sum_{j=1}^{J} R_{j,3} = O(J)$.

5. Complexity of the algorithm and an acceleration scheme

The size of the state space of the Markov chain in Algorithm 1 is $(2|A|)^N$. For large N this is huge, e.g., if $N = 16$ and there are two actions, the state space size is 2^{32}. This makes the convergence to the stationary distribution of the perturbed Markov chain very slow. This makes Algorithm 1 unsuitable for large values of M, N. In this section, we propose an algorithm that significantly reduces the running time of Algorithm 1. The algorithm that we discuss in this section exploits the structure of the Markov chain, to base the agents' decision on a much smaller subset of the state space. This results in faster convergence to the conditional stationary distribution.

As the main theorem states for sufficiently small ε the Markov chain concentrates with high probability in the all-content state and the optimal action $\left(C^N, \mathbf{a}^*\right)$. The mixing time of this Markov chain can be large. Although we have no explicit bound for the mixing time, it can grow as $O(M^N)$, where $M = |A|$ is the cardinality of the action set and N is the number of agents.

We would like to provide some insight into the dynamics which will allow us to develop a heuristic that converges much faster. Each agent acts independently. Hence the probability of an agent becoming content at time t when it is discontent is $\varepsilon^{U_{\max} - U_n(t)}$. Hence (assuming that $U_{max} = 1$), this is larger than ε. Therefore, it is expected that using random actions the agent will become content in time shorter than ε^{-1}. On the other hand, once it is content it will change its action and therefore become discontent is ε^c, and $c > N$. Hence after time shorter than $\beta N \varepsilon^{-1}$ for any $\beta > 1$ all users will become content. All the agents will remain content for time interval $(1 - N\varepsilon^c)^{-1}$ which can be approximated by $\frac{1}{N\varepsilon^c}$ which is much longer. Hence we expect that most of the time the Markov chain will be in an all-content state. Conditioning on an all-content state, the probability that the transition was using an action vector $\mathbf{a}$ is proportional to $\varepsilon^{NU_{\max} - \sum_{n=1}^{N} U_n(\mathbf{a})}$. Specifically,

$$P(\mathbf{a}) = \frac{\varepsilon^{NU_{\max} - \sum_{n=1}^{N} U_n(\mathbf{a})}}{\sum_{\mathbf{a} \in A^N} \sum_{N=1}^{N} \varepsilon^{NU_{\max} - \sum_{n=1}^{N} U_n(\mathbf{a})}}. \tag{25}$$

So the all-content states appear with a relatively high probability. If the agents could identify these all-content states, they could choose an action based on the frequency of each action in the all-content states. This would become for each agent a problem of finding the maximum of $|A|$ counting processes, each with transition probability $P(\mathbf{a}) = \varepsilon^{U_{\max} - U_n(\mathbf{a})}$. A standard concentration argument shows that with probability 1 and exponentially fast, the correct state will be chosen except finitely many cases in this random walk. This is the insight behind the analysis of the stationary probability. However, this insight suggests that there is a significant value in identifying the all-content state, since this will accelerate the convergence to the optimal action.

Following the above discussion, we would like to identify the states where all players are content, and only take into account these states when determining the action. Surprisingly, this can be done distributedly, by adding to the negotiation phase, another sub-phase. In this sub-phase, all users apply action 1 for $k^{1+\delta/10}$ steps. Following this, all the content users apply action 1 again, while discontent users do not act for another $k^{1+\delta/10}$ steps. If some users are discontent, the value for each content user during the second sub-phase is larger than in the first sub-phase by at least $\frac{U_{n,1}}{N(N-1)}$, since the load in this phase is reduced. As k increases the probability of misidentifying the all content state by any user approaches 0 exponentially fast, using standard concentration arguments. This amounts to adding Algorithm 2 at the end of each block j in the negotiation phase after line 27 of Algorithm 1

$V(a)$ is a vector of counters that counts how many times the agent visited a in an all-content state. It is updated once every I blocks. Now, Eq. (21) is replaced with

$$m_n^{j,3} = \arg\max_a V(a). \tag{26}$$

6. Numerical examples

We illustrate the performance of the ENE algorithm in the non-asymptotic regime in comparison with a random allocation and the distributed (selfish) Upper Confidence Bound (dUCB) algorithm (Besson & Kaufmann, 2018).

Suppose $N = 4$ agents share $M = 2$ communication channels. When two agents or more choose the same channel at the

Algorithm 2 Identify all-content states

Input: $V(a) : a \in A$, Mood, $a_n = m_n^{j,2,k}$.
Let $I = \lfloor \frac{1}{\varepsilon} \rfloor$.
Select $a = 1$.
$v_{n,N} = 0$.
for $i = 1$ **to** $j^{1+\delta/10}$ **do**
  Perform action 1 and store $r^1_{n,N}$.
  $v_{n,N} \leftarrow \frac{i-1}{i} v_{n,N} + \frac{1}{i} r^1_{n,N}$.
end for
if mood=C **then**
  Select $a = 1$.
  $v_{n,N-1} = 0$.
  for $i = 1$ **to** $j^{1+\delta/10}$ **do**
    Perform action 1 and store $r^1_{n,N-1}$.
    $v_{n,N-1} \leftarrow \frac{i-1}{i} v_{n,N-1} + \frac{1}{i} r^1_{n,N-1}$
  end for
  if $v_{n,N-1} > v_{n,N} \left(1 + \frac{1}{2N(N-1)}\right)$ **then**
    mood= D.
  else
    mood $\leftarrow C$.
    All content $\leftarrow 1$.
    $V(a_n) = V(a_n) + 1$.
  end if
end if

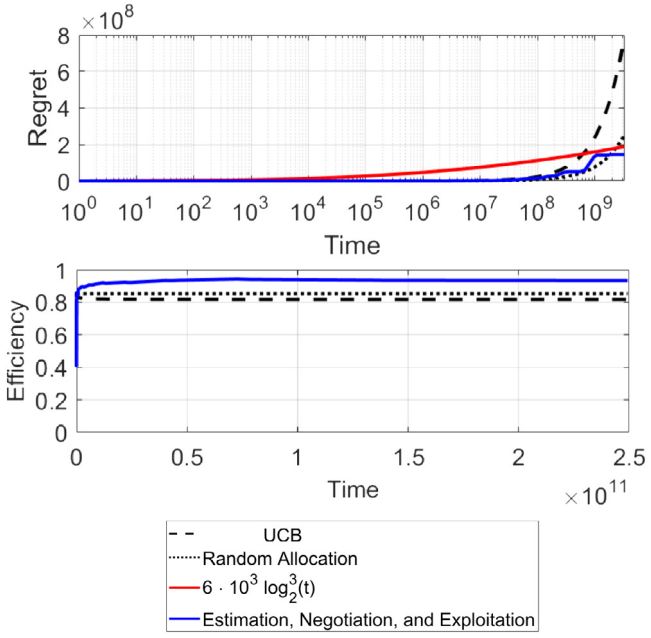

Fig. 2. Comparing the ENE algorithm, the dUCB, and a random allocation over time. (a) Regret for the matrix (27). (b) Average Efficiency for 50 random matrices (30).

same time, they share it equally via a round-robin Time Division Multiple Access (TDMA) mechanism. Assume one communication channel has high throughput (strongly desirable) and the other has low throughput (weakly desirable). If all agents use the high throughout the channel, then the channel becomes overcrowded and its throughout reduces. We illustrate how the agents can achieve an optimal distributed allocation over the channels using the ENE algorithm.

To improve the convergence speed the number of blocks and their duration can be scaled by fixed constants to better fit the specific meta-parameters which represent the initial number of iterations in each phase (c_1, c_2, c_3) This is a feature of the ENE algorithm and does not change the $O(\log_2^3(T))$ regret guarantee. In the first set of simulations the noise ν was Gaussian with variance 0.1, the constant α is 1, the constant c is N, and $\varepsilon = 10^{-2}$. We set the parameter $\delta = 1$. We used $c_1 = 5 \cdot 10^2$, $c_2 = 5 \cdot 10^3$, $c_3 = 2.5 \cdot 10^6$. The number of epochs was 10. The matrices $U_{n,m,1}$ were:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0.24 & 0.24 & 0.24 & 0.24 \end{pmatrix} \tag{27}$$

Fig. 2(a) depicts the sample path of the regret when It can be seen that the regret is indeed $O(\log_2^3(t))$, while the two other algorithms suffer a linear regret.

In the second set of simulations, we estimated the efficiency of the algorithm, defined by:

$$\text{Efficiency} \triangleq \frac{\left(\frac{1}{T}\sum_{t=1}^{T} W^t\right) - W_{\text{worst}}}{W^* - W_{\text{worst}}} \tag{28}$$

where the worst welfare is:

$$W_{\text{worst}} \triangleq \min_{\mathbf{m}} W(\mathbf{m}) \tag{29}$$

Fig. 2(b) depicts the average efficiency of 50 random trials. The matrices $U_{n,m,1}$ are:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0.2 & 0.2 & 0.2 & 0.2 \end{pmatrix} - Q, \tag{30}$$

where Q is a random matrix where all the entries are uniformly distributed between 0 and 0.2. The steady-state efficiency of the ENE algorithm is 93% while for the random allocation, it is 85% and for the selfish UCB it is only 82%.

High dimension example. We simulated the following example: $N = 16, M = 2$. In this case, the joint state space is of size 2^{32}. However, as we will show, using the modified protocol (Algorithm 2 of Section 5) results in faster convergence, and the exploitation regret becomes 0 relatively quickly. As mentioned, the methodology of Bistritz et al. (2021) and Bistritz and Leshem (2020) provides a finer analysis of the Markov chain, showing that we can use previous epochs to estimate the most frequent state. Hence we used a negotiation phase of length $c_2 j^{0.25}$ during epoch j where $\delta = 0.75$, instead of $1 + 0.25$, and the current state utility estimation used $2j^{1.25}$ samples. This resulted in $O\left(\log^{2.5}(T)\right)$ regret. We selected a random non-negative 16×2 reward matrix with a difference between utilities in different states of each player constrained to be larger than 0.2. A standard $N(0, 1)$ Gaussian noise was added to each measurement. The parameters of the simulation were $\varepsilon = 0.001, c_1 = 10^5, c_2 = 10^5$, $c_3 = 10^3$, $c = 1.1$. We performed 200 independent Monte-Carlo simulations of 24 epochs each. The results are displayed in Fig. 3. We can clearly see the nearly linear dependence of the regret on $\log(T)$ as expected. The figure also displays the 10% and 90% confidence bounds on the regret. We can see that even though the 90% regret curve is higher, it is within the ballpark of the median regret curve and not an order of magnitude larger.

Finally, we comment on the choice of c_1, c_2, c_3. Estimation of the utilities during exploration is faster than the convergence time of the Markov chain. Furthermore, estimating the load requires a large number of samples per iteration, so c_1 can be chosen smaller or equal to c_2 without significant impact on the regret. On the other hand, following the discussion of the Markov chain convergence in Section 5, we require that c_2 is significantly larger than ε^{-1}. Finally, the choice of c_3 does not significantly impact the exploitation regret since the duration of the exploitation phase grows exponentially with the number of epochs.

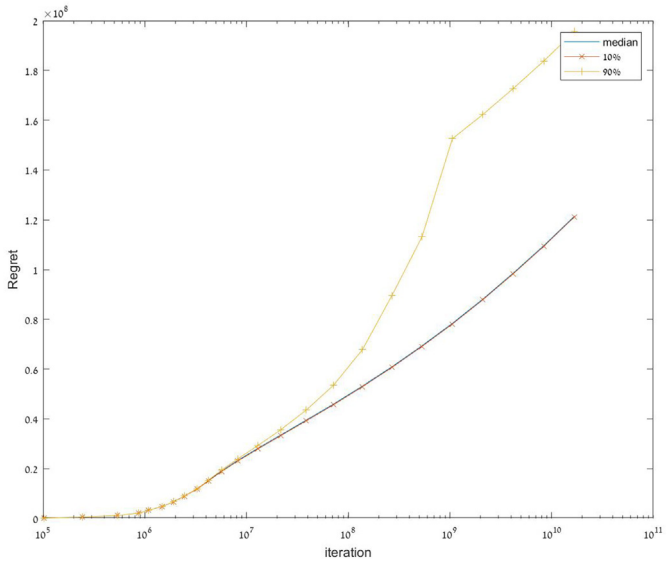

Fig. 3. Regret vs. time, with Algorithm 2 for $N = 16$, $M = 2$. Also shown are the 10% and 90% confidence bounds $+$, $\times$ respectively. Also the median regret is shown since it disregards outliers.

7. Conclusions

This paper extends the multi-agent multi-armed bandit problem formulation and the regret analysis to the heavily congested case where multiple agents can share a resource without any communication between agents. The problem is motivated by applications to cloud computing, spectrum collaboration (in wireless networks), and machine scheduling (in operations research). Our formulation uses a congestion game with incomplete information. We propose a novel three-phased algorithm with provable expected regret of $O\left(\log_2^{3+\delta}(T)\right)$. In contrast to prior art we propose to add an identification of the all-content state to accelerate the convergence. This allows us to implement the algorithm even for 16 agents and state space of size 2^{32} with very good results. Simulation results show that regret order can be made lower.

Appendix

A.1. Properties of sub-Gaussian random variables

Let ν be a sub-Gaussian random variable with variance proxy b. The moment-generating function of ν is bounded by

$$\mathbb{E}(e^{\nu s}) \leq \exp\left(\frac{bs^2}{2}\right) \quad \forall s \in \mathbb{R} \tag{A.1}$$

The following properties are repeatedly used in the proofs of the lemmas of the main paper. For proofs, see e.g., Wainwright (2019).

Lemma A.1. *Let ν be a sub-Gaussian random variable with variance proxy b. Let g be some constant. The random variable $g\nu$ is sub-Gaussian with variance proxy bg^2.*

Lemma A.2. *The average of k independently and identically distributed sub-Gaussian random variables with variance proxy b is sub-Gaussian with variance proxy b/k.*

An immediate corollary of Lemmas A.1 and A.2 is that the average of k independently and identically distributed sub-Gaussian random variables with variance proxy b times a constant g is a sub-Gaussian random variable with variance proxy $\frac{bg^2}{k}$:

$$\mathbb{E}\left(\exp\left(gs\frac{1}{k}\sum_{i=1}^{k}\nu_i\right)\right) \leq \exp\left(\frac{bs^2g^2}{2k}\right) \tag{A.2}$$

Lemma A.3. *Let ν_1 and ν_2 be independent sub-Gaussian random variables with variance proxies b_1 and b_2 respectively, then $\nu_1 + \nu_2$ is sub-Gaussian with variance proxy $b_1 + b_2$.*

Lemma A.4. *Let ν be sub-Gaussian random variable with variance proxy b. For any s, it holds that*

$$\mathbb{P}(\nu > s) \leq e^{-\frac{s^2}{2b}} \qquad \text{and} \qquad \mathbb{P}(\nu < -s) \leq e^{-\frac{s^2}{2b}} \tag{A.3}$$

An immediate consequence of Lemma A.4 and Eq. (A.2) is that the probability that a constant g times the absolute value of the average of k independently and identically distributed sub-Gaussian random variables with variance proxy b will be greater than s is bounded as follows:

$$\mathbb{P}\left(\left|\frac{g}{k}\sum_{i=1}^{k}\nu_i\right| > s\right) \leq 2\exp\left(-\frac{ks^2}{2bg^2}\right). \tag{A.4}$$

A.2. Proof of (10)

If agent n was active in time slot τ of block $M+1$ of phase 1 of epoch j its load is a random variable distributed as follows:

$$\ell_n^{j,1,M+1,\tau} \sim \text{Binomial}(N-1, 1/2) + 1 \tag{A.5}$$

Let us consider the more general case where the probability of an agent to be active in time slot τ of block $M+1$ of phase 1 of epoch j is not 1/2 but rather p. The first inverse moment of this random variable can be obtained as follows:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{\ell_n^{j,1,M+1,\tau}}\right] &= \sum_{i=0}^{N-1}\frac{1}{i+1}\binom{N-1}{i}p^i(1-p)^{N-1-i} \\
&= \sum_{i=0}^{N-1}\frac{1}{N}\binom{N}{i+1}p^i(1-p)^{N-1-i} \\
&= \frac{1}{pN}\sum_{i=0}^{N-1}\binom{N}{i+1}p^{i+1}(1-p)^{N-1-i} \\
&= \frac{1}{pN}\sum_{i=0}^{N-1}\binom{N}{i+1}p^{i+1}(1-p)^{N-1-i} \\
&\quad + \frac{(1-p)^N}{pN} - \frac{(1-p)^N}{pN} \\
&= \frac{1}{pN}\sum_{i=0}^{N}\binom{N}{i}p^i(1-p)^{N-i} - \frac{(1-p)^N}{pN} \\
&= \frac{1}{pN} - \frac{(1-p)^N}{pN} = \frac{1-(1-p)^N}{pN}
\end{aligned} \tag{A.6}$$

Setting $p = 1/2$ completes the proof.

A.3. Lemmas in proof of main theorem

Lemma A.5. $\mathbb{P}(E^{j,3}) = O(\exp(-j^{1+\delta/4}))$.

Proof. Estimation and Negotiation Phase failures are denoted by

$$E^{j,1} : \mathbf{m}^{*j} \neq \mathbf{m}^* \tag{A.7}$$

$$E^{j,2} : \mathbf{m}^{j,3} \neq \mathbf{m}^{*j} \tag{A.8}$$

The probabilities of these are bounded by $O(\exp(-j^{1.4}))$ and $O(\exp(-j^{1+\delta/4}))$, respectively, according to Lemmas A.6 and A.9, respectively. Furthermore,

$$\mathbb{P}(E^{j,3}) \leq \mathbb{P}(E^{j,1}) + \mathbb{P}(E^{j,2}) \tag{A.9}$$

Lemma A.6. $\mathbb{P}(E^{j,1}) = O\left(\exp\left(-j^{1.4}\right)\right)$.

Proof. Let us define the following errors events:

$$\tilde{E}^{j,1} : \max_{n,m,\ell} \left| U_{n,m,\ell} - \hat{U}^{j}_{n,m,\ell} \right| \geq \rho \tag{A.10}$$

$$\tilde{E}^{j,1}_{N} : \exists n : \hat{N}^{j}_{n} \neq N \tag{A.11}$$

$$\tilde{E}^{j,1}_{U} : \max_{n,m} \left| U_{n,m,N} - \hat{U}^{j}_{n,m,N} \right| > \frac{\rho}{N} \tag{A.12}$$

We bound $\mathbb{P}\left(E^{j,1}\right)$ as follows:

$$\mathbb{P}(E^{j,1}) \overset{(a)}{\leq} \mathbb{P}\left(\tilde{E}^{j,1}\right) \overset{(b)}{\leq} \mathbb{P}\left(\tilde{E}^{j,1}_{N}\right) + \mathbb{P}\left(\tilde{E}^{j,1}_{U}\right) \tag{A.13}$$

where (a) is a simple modification of Lemma (1) in Bistritz and Leshem (2018a), and (b) is clear from (12). The probability $\mathbb{P}\left(\tilde{E}^{j,1}_{N}\right)$ is $O(\exp(-j^{1.4}))$ according to Lemma A.7. We bound $\mathbb{P}\left(\tilde{E}^{j,1}_{U}\right)$ as follows:

$$\begin{aligned}
\mathbb{P}\left(\tilde{E}^{j,1}_{U}\right) &\overset{(a)}{\leq} NM\mathbb{P}\left(\left|U_{n,m,N} - \hat{U}^{j}_{n,m,N}\right| > \frac{\rho}{N}\right) \\
&\overset{(b)}{\leq} 2NM \exp\left(-\frac{1}{2b}\left(\frac{\rho}{N}\right)^{2}\frac{j^2+j}{2}\right) \\
&\overset{(c)}{\leq} 2NM \exp\left(-\frac{j^2}{4b}\left(\frac{\rho}{N}\right)^{2}\right) = O(\exp(-j^{1.5}))
\end{aligned} \tag{A.14}$$

where (a) is a union bound on the agents and resources, (b) is Chernoff's inequality for the average of i.i.d. sub-Gaussian random variables, and (c) holds for any positive j.

Lemma A.7. $\mathbb{P}\left(\tilde{E}^{j,1}_{N}\right) = O(\exp(-j^{-1.4}))$.

Proof. An error event by agent n in block k of phase 1 of epoch j is denoted with $E^{j,1,k}_{n}$ and defined by

$$\left|\bar{r}^{j,1,k}_{n} - \mathbb{E}\left[\bar{r}^{j,1,k}_{n}\right]\right| > 2^{-N-4}\mathbb{E}\left[\bar{r}^{j,1,k}_{n}\right] \tag{A.15}$$

As is proved in appendix A.7 the following holds:

$$\mathbb{P}\left(\hat{N}^{j}_{n} \neq N\right) \leq \mathbb{P}\left(E^{j,1,1}_{n}\right) + \mathbb{P}\left(E^{j,1,M+1}_{n}\right) \tag{A.16}$$

According to Chernoff's inequality:

$$\mathbb{P}\left(E^{j,1,1}_{n}\right) \leq 2\exp\left(-\frac{j^2}{4b}\left(U_{n,1,N}\cdot 2^{-N-4}\right)^{2}\right) \tag{A.17}$$

According to Lemma A.8 the probability $\mathbb{P}\left(E^{j,1,M+1}_{n}\right)$ is $O(\exp(-j^{1.4}))$. A union bound on the agents preserves the asymptotic behavior in j such that $\mathbb{P}\left(\tilde{E}^{j,1}_{N}\right)$ is upper bounded by the same order.

Lemma A.8. $\mathbb{P}(E^{j,1,M+1}_{n}) = O(\exp(-j^{1.4}))$

Proof. Let the number of samples collected by agent n during all epochs until epoch j in phase 1 and block $M+1$ be:

$$\xi^{j,1,M+1}_{n} \triangleq \sum_{i=1}^{j}\sum_{\tau=1}^{i} \mathbb{1}\left(m^{i,1,M+1,\tau}_{n} = 1\right) \tag{A.18}$$

The probability $\mathbb{P}\left(E^{j,1,M+1}_{n}\right)$ is upper bounded by the sum of the following two probabilities:

$$\mathbb{P}\left(\xi^{j,1,M+1}_{n} < \frac{j^{1.5}}{2}\right) \tag{A.19}$$

$$\mathbb{P}\left(E^{j,1,M+1}_{n} \,\middle|\, \xi^{j,1,M+1}_{n} > \frac{j^{1.5}}{2}\right) \tag{A.20}$$

Since $\xi^{j,1,M+1}_{n}$ is a binomial random variable with parameters $\frac{1}{2}(j^2+j)$ and $1/2$, the probability in (A.19) can be upper bounded with Hoeffding's inequality by the following expression:

$$\begin{aligned}
&\exp\left(-2\cdot\frac{1}{2}(j^2+j)\left(\frac{1}{2} - \frac{j^{1.5}}{2\cdot\frac{1}{2}(j^2+j)}\right)^{2}\right) \\
&\overset{(a)}{<} \exp\left(-j^2\left(\frac{1}{2} - \frac{1}{j^{0.5}}\right)^{2}\right) \overset{(b)}{<} \exp\left(\frac{-j^2}{100}\right)
\end{aligned} \tag{A.21}$$

where (a) holds for any epoch and (b) holds from the seventh epoch. This is of course $O(\exp(-j^{1.4}))$.

Since $\bar{r}^{j,1,M+1}_{n}$ is the sum of a sub-Gaussian random variable with variance proxy b and another random variable bounded between $U_{n,1,1}$ and $U_{n,1,N}$ then the probability in (A.20) can be upper bounded with Chernoff–Hoeffding's inequality by the following expression:

$$2\exp\left(-\frac{1}{4}\cdot\frac{\left(\mathbb{E}\left[\bar{r}^{j,1,M+1}_{n}\right]\cdot 2^{-N-4}\right)^{2} j^{1.5}}{b + (U_{n,1,1} - U_{n,1,N})^2}\right) \tag{A.22}$$

The last expression is also $O(\exp(-j^{1.4}))$.

Lemma A.9. $\mathbb{P}(E^{j,2}) = O\left(\exp\left(-j^{1+\delta/4}\right)\right)$.

Proof. A Load Estimation Error occurs when at least one agent incorrectly estimates its load during at least one block of the Negotiation Phase:

$$E^{j,2}_{\text{load}} : \exists n, k : \hat{\ell}^{j,2,k}_{n} \neq \ell^{j,2,k}_{n} \tag{A.23}$$

The probability of this error is $O\left(\exp\left(-j^{1+\delta/4}\right)\right)$ according to Lemma A.10.

An Insufficient Mixing Time Error is defined as follows:

$$E^{j,2}_{\text{mix}} = E^{j,2} \wedge \neg E^{j,2}_{\text{load}} \tag{A.24}$$

The probability of this error is $O\left(\exp\left(-j^{1+\delta/4}\right)\right)$ according to Lemma A.13.

To finish this lemma:

$$\mathbb{P}\left(E^{j,2}\right) \leq \mathbb{P}\left(E^{j,2}_{\text{load}}\right) + \mathbb{P}\left(E^{j,2}_{\text{mix}}\right). \tag{A.25}$$

Lemma A.10. $\mathbb{P}\left(E^{j,2}_{load}\right) = O\left(\exp\left(-j^{1+\delta/4}\right)\right)$

Proof. Agent n will incorrectly estimate its load in block k if the average reward it receives in that block with resource $m^{j,2,k}_{n} = m$ and load $\ell^{j,2,k}_{n} = \ell$ is sufficiently far from its estimated utility:

$$\left| r^{j,2,k}_{n,m,\ell} - \hat{U}^{j}_{n,m,\ell} \right| > \Phi \tag{A.26}$$

where Φ is defined as

$$\Phi \triangleq \frac{1}{3}\min_{n,m}\left|\hat{U}^{j}_{n,m,N-1} - \hat{U}^{j}_{n,m,N}\right| \tag{A.27}$$

The difference between the average reward and estimated utility in the left-hand side of (A.26) is a sub-Gaussian random variable with variance proxy $\frac{b}{j^{1+\delta/3}} + \frac{2b}{j^2+j} = b\frac{j+2j^{\delta/3}+1}{j^{2+\delta/3}+j^{\delta/3+1}} < \frac{2b}{j^{1+\delta/3}}$. A union bound on (A.26) with respect to the agents and

load-estimation-blocks together with Chernoff's bound produces:

$$\mathbb{P}\left(E^{j,2}_{\text{load}}\right) \leq 2Nj^{1+\delta/3}\exp\left(-\frac{\Phi^2 j^{1+\delta/3}}{2b}\right) = O\left(\exp\left(-j^{1+\delta/4}\right)\right) \tag{A.28}$$

Lemma A.11. $\mathbb{P}\left(E^{j,2}_{mix}\right) = O\left(\exp\left(-j^{1+\delta/4}\right)\right)$

Proof. Let $\mathcal{M}^j$ be the Markov chain of the Negotiation Phase of epoch j. Define the optimal state to be the estimated optimal allocation (13) with the optimal estimated utilities (14) and all agents Content:

$$z^* \triangleq (\mathbf{m}^{*j}, \mathbf{U}^{*j}, C^N). \tag{A.29}$$

Let $\tilde{\mathcal{M}}^j$ be a Markov Chain on $\mathcal{Z}$ identical to $\mathcal{M}^j$ except that a Load Estimation Error is impossible. The stationary probability in $\tilde{\mathcal{M}}^j$ of the optimal state is π^*. According to Lemma A.12: $\pi^* > 2/3$. The state of the Markov Chain in block k of the Negotiation Phase of epoch j is $z^{j,2,k}$. The estimated stationary probability of the optimal state is:

$$\hat{\pi}^* \triangleq \frac{1}{\alpha j^{1+\delta/3}} \sum_{k=(1-\alpha)j^{1+\delta/3}}^{j^{1+\delta/3}} \mathbb{1}(z^{j,2,k} = z^*) \tag{A.30}$$

We wish to bound the probability that the optimal state was visited in less than half the blocks of the tail of the Negotiation Phase due to insufficient mixing time of the Markov chain. According to Lemma A.13 this is:

$$\mathbb{P}\left(\hat{\pi}^* \leq \frac{1}{2} \leq \pi^* \frac{3}{4}\right) = O\left(\exp\left(-j^{1+\delta/4}\right)\right) \tag{A.31}$$

Lemma A.12. $\pi^* > 2/3$

Proof. Any two agents can increase or decrease each other's utilities by choosing the same action or a different action, respectively. Formally, this is known as interdependence and is defined in Marden et al. (2014). As $\varepsilon \to 0$ we have $\pi^* \to 1$. This is due to the interdependence of our dynamics and Theorem 3.2 from Marden et al. (2014). For our ε, the stationary probability of the optimal state is $\pi^* > 2/3$.

Lemma A.13. $\mathbb{P}\left(\hat{\pi}^* \leq \frac{1}{2} \leq \pi^* \frac{3}{4}\right) = O\left(e^{-j^{1+\delta/4}}\right)$

Proof. Let $\mathcal{Z}$ be the set of states of $\tilde{\mathcal{M}}^j$ and Let π be its stationary distribution. Let φ be the distribution of block $(1-\alpha)j^{1+\delta/3}$. Let $\|\varphi\|_\pi$ be the π-norm of φ defined by:

$$\|\varphi\|_\pi \triangleq \sqrt{\sum_{z\in\mathcal{Z}} \frac{\varphi^2(z)}{\pi(z)}} \tag{A.32}$$

Let $T_{1/8}$ be the minimal amount of time necessary for $\tilde{\mathcal{M}}^j$ to reach a total variation distance of $1/8$ from π with arbitrary initialization. Let C be a positive constant, independent of φ, $T_{1/8}$, and π. Let $\eta \triangleq 1 - \frac{1}{2\pi^*}$. According to Lemma A.12 we have $\pi^* > 2/3$ and therefore $\eta < 1$. After setting $\frac{1}{2} = (1-\eta)\pi^*$ into (A.31), the Markovian concentration bound from Chung et al. (2012) produces the following upper bound on the probability of an Insufficient Mixing

Time Error of (A.31):

$$C\|\varphi\|_\pi \exp\left(-\frac{\left(1-\frac{1}{2\pi^*}\right)^2 \pi^* \alpha j^{1+\delta/3}}{72 T_{1/8}}\right) \tag{A.33}$$

Because $\pi^* > 2/3$ we obtain

$$\left(1-\frac{1}{2\pi^*}\right)^2 \pi^* \geq \frac{1}{24} \tag{A.34}$$

setting this into (A.33) completes the proof.

There is a tradeoff regarding $\|\varphi\|_\pi$. Starting from an arbitrary initial condition, $\|\varphi\|_\pi$ can be large. By dedicating the first $\alpha j^{1+\delta/3}$ blocks of the Negotiation Phase to letting the Markov chain approach its stationary distribution, and starting to count the visits to z^* only afterward, we can reduce $\|\varphi\|_\pi$ significantly, at the cost of $(1-\alpha)j^{1+\delta/3}$ turns less for estimating z^*. Optimizing over α can improve the constants of the bound in (A.33).

A.4. Proof of Eq. (A.16)

Agent n will correctly estimate the number of agents when

$$N - \frac{1}{2} < \frac{1}{\ln(1/2)} \ln\left(1 - \frac{\bar{r}_n^{j,1,M+1}}{2\bar{r}_n^{j,1,1}}\right) < N + \frac{1}{2} \tag{A.35}$$

After applying some algebra to (A.35) we obtain:

$$2\left(1-\frac{1}{2^N}\right) + \frac{1}{2^{N-1}}\left(1-\sqrt{2}\right) \leq \frac{\bar{r}_n^{j,1,M+1}}{\bar{r}_n^{j,1,1}} \leq 2\left(1-\frac{1}{2^N}\right) + \frac{1}{2^{N-1}}\left(1-\frac{1}{\sqrt{2}}\right) \tag{A.36}$$

Because $\left|1-2^{1/2}\right| > 1-2^{-1/2}$ we can make the lower bound in (A.36) tighter:

$$2\left(1-\frac{1}{2^N}\right) - \frac{1}{2^{N-1}}\left(1-\frac{1}{\sqrt{2}}\right) \leq \frac{\bar{r}_n^{j,1,M+1}}{\bar{r}_n^{j,1,1}} \leq 2\left(1-\frac{1}{2^N}\right) + \frac{1}{2^{N-1}}\left(1-\frac{1}{\sqrt{2}}\right) \tag{A.37}$$

We now wish to move from an additive bound to a multiplicative bound. We take notice of the following:

$$\frac{2^{1-N}(1-2^{-1/2})}{2(1-2^{-N})} > \frac{(1-2^{-1/2})}{2^N} > \frac{1/4}{2^N} > 2^{-2-N}, \tag{A.38}$$

We use (A.38) to make the bounds in (A.37) tighter:

$$2\left(1-\frac{1}{2^N}\right)\left(1-2^{-N-2}\right) \leq \frac{\bar{r}_n^{j,1,M+1}}{\bar{r}_n^{j,1,1}} \leq 2\left(1-\frac{1}{2^N}\right)\left(1+2^{-N-2}\right) \tag{A.39}$$

We multiply all sides of (A.39) by the ratio of the expectations:

$$\left(1-2^{-N-2}\right) \leq \frac{\bar{r}_n^{j,1,M+1}}{\bar{r}_n^{j,1,1}} \cdot \frac{\mathbb{E}\left[\bar{r}_n^{j,1,1}\right]}{\mathbb{E}\left[\bar{r}_n^{j,1,M+1}\right]} \leq \left(1+2^{-N-2}\right) \tag{A.40}$$

The last expression will hold if

$$\sqrt{\left(1-2^{-N-2}\right)} \leq \frac{\bar{r}_n^{j,1,M+1}}{\mathbb{E}\left[\bar{r}_n^{j,1,M+1}\right]} \leq \sqrt{\left(1+2^{-N-2}\right)} \tag{A.41}$$

$$\sqrt{\left(1-2^{-N-2}\right)} \leq \frac{\mathbb{E}\left[\bar{r}_n^{j,1,1}\right]}{\bar{r}_n^{j,1,1}} \leq \sqrt{\left(1+2^{-N-2}\right)} \tag{A.42}$$

Subtracting 1 from all sides of (A.41) and (A.42) and rearranging produces:

$$\mathbb{E}\left[\bar{r}_n^{j,1,M+1}\right]\left(\sqrt{\left(1-2^{-N-2}\right)} - 1\right) \leq \bar{r}_n^{j,1,M+1} - \mathbb{E}\left[\bar{r}_n^{j,1,M+1}\right] \leq \mathbb{E}\left[\bar{r}_n^{j,1,M+1}\right]\left(\sqrt{\left(1+2^{-N-2}\right)} - 1\right) \tag{A.43}$$

$$\mathbb{E}\left[\bar{r}_n^{j,1,1}\right]\left(\frac{1}{\sqrt{\left(1-2^{-N-2}\right)}}-1\right)\geq \bar{r}_n^{j,1,1}-\mathbb{E}\left[\bar{r}_n^{j,1,1}\right]\geq \mathbb{E}\left[\bar{r}_n^{j,1,1}\right]\left(\frac{1}{\sqrt{\left(1+2^{-N-2}\right)}}-1\right) \tag{A.44}$$

For convenience we define the following bounds and notice that the upper bound is tighter in (A.43) while the lower bound is tighter in (A.44):

$$L_1 \triangleq \left|\frac{1}{\sqrt{\left(1+2^{-N-2}\right)}}-1\right| \leq \frac{1}{\sqrt{\left(1-2^{-N-2}\right)}}-1 \tag{A.45}$$

$$L_2 \triangleq \sqrt{\left(1+2^{-N-2}\right)}-1 \leq \left|\sqrt{\left(1-2^{-N-2}\right)}-1\right| \tag{A.46}$$

We make the bounds in (A.43) and (A.44) tighter according to our observations from (A.45) and (A.46):

$$\left|\bar{r}_n^{j,1,1}-\mathbb{E}\left[\bar{r}_n^{j,1,1}\right]\right| \leq \mathbb{E}\left[\bar{r}_n^{j,1,1}\right] L_1 \tag{A.47}$$

$$\left|\bar{r}_n^{j,1,M+11}-\mathbb{E}\left[\bar{r}_n^{j,1,M+1}\right]\right| \leq \mathbb{E}\left[\bar{r}_n^{j,1,M+1}\right] L_2 \tag{A.48}$$

We use Taylor's series to obtain the following bounds:

$$\frac{1}{\sqrt{1+x}} \leq 1-0.5x+0.375x^2 \tag{A.49}$$

$$\sqrt{1+x} \geq 1+x/2-x^2/8 \tag{A.50}$$

We use (A.49) and (A.50) to obtain the following bounds:

$$L_1 \geq 1-1+2^{-N-3}-0.375\cdot 2^{-2N-4} \geq 2^{-N-4} \tag{A.51}$$

$$L_2 \geq 1+2^{-N-3}-2^{-2N-7}-1 \geq 2^{-N-4} \tag{A.52}$$

When (A.51) and (A.52) hold, then (A.35) holds.

A.5. Proof of (22)

The proof of (22) is quite straightforward. We need to prove that

$$\sum_{j=1}^{J} j^{2+2\delta/3} = O(J^{3+\delta}). \tag{A.53}$$

To do so we begin with the following equation:

$$(j^{1+\delta/3}-1)^3 = j^{3+\delta}-3j^{2+2\delta/3}+3j^{1+\delta/3}-1 \tag{A.54}$$

Rearrange (A.54) to obtain:

$$3j^{2+2\delta/3}-3j^{1+\delta/3}+1 = j^{3+\delta}-(j^{1+\delta/3}-1)^3 \leq j^{3+\delta}-(j-1)^{3+\delta} \tag{A.55}$$

We take (A.55) and sum over all the epochs of the algorithm :

$$\sum_{j=1}^{J} 3j^{2+2\delta/3}-3j^{1+\delta/3}+1 \leq \sum_{j=1}^{J} j^{3+\delta}-(j-1)^{3+\delta} = J^{3+\delta} \tag{A.56}$$

We rearrange (A.56) to obtain

$$\sum_{j=1}^{J} j^{2+2\delta/3} \leq \frac{J^{3+\delta}-J}{3}+\sum_{j=1}^{J} j^{1+\delta/3} = O(J^{3+\delta}) \tag{A.57}$$

References

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

Avner, O., & Mannor, S. (2019). Multi-user communication networks: A coordinated multi-armed bandit approach. *IEEE/ACM Transactions on Networking*, *27*(6), 2192–2207.

Bertsekas, D. P. (1979). *A distributed algorithm for the assignment problem: Lab. for Information and Decision Systems Working Paper*, MIT.

Besson, L., & Kaufmann, E. (2018). Multi-player bandits revisited. In *Algorithmic learning theory* (pp. 56–92). PMLR.

Bistritz, I., Baharav, T. Z., Leshem, A., & Bambos, N. (2021). One for all and all for one: Distributed learning of fair allocations with multi-player bandits. *IEEE Journal on Selected Areas in Information Theory*, *2*(2), 584–598.

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

Bistritz, I., & Leshem, A. (2018b). Game of thrones: Fully distributed learning for multi-player bandits. ArXiv preprint arXiv:1810.11162.

Bistritz, I., & Leshem, A. (2019). Game theoretic dynamic channel allocation for frequency-selective interference channels. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, *65*(1), 330–353.

Bistritz, I., & Leshem, A. (2020). Game of thrones: Fully distributed learning for multi-player bandits. *Mathematics of Operations*.

Blumrosen, L., & Dobzinski, S. (2007). Welfare maximization in congestion games. *IEEE Journal on Selected Areas in Communications*, *25*(6), 1224–1236.

Boursier, E., Perchet, V., Kaufmann, E., & Mehrabian, A. (2019). A practical algorithm for multiplayer bandits when arm means vary among players. arXiv preprint arXiv:1902.01239.

Boyarski, T., Wang, W., & Leshem, A. (2023). Distributed learning for optimal spectrum access in dense device-to-device ad-hoc networks. *IEEE Transactions on Signal Processing*.

Bubeck, S., Li, Y., Peres, Y., & Sellke, M. (2019). Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without. arXiv preprint arXiv:1904.12233.

Cheng, N., Zhang, N., Lu, N., Shen, X., Mark, J. W., & Liu, F. (2013). Opportunistic spectrum access for CR-VANETs: A game-theoretic approach. *IEEE Transactions on Vehicular Technology*, *63*(1), 237–251.

Chung, K. M., Lam, H., L, Z., & Mitzenmacher, M. (2012). Chernoff-Hoeffding bounds for Markov chains: Generalized and simplified. In *29th international symposium on theoretical aspects of computer science* (p. 124).

Darak, S. J., & Hanawal, M. K. (2019). Multi-player multi-armed bandits for stable allocation in heterogeneous ad-hoc networks. *IEEE Journal on Selected Areas in Communications*, *37*(10), 2350–2363.

Kalathil, D., Nayyar, N., & Jain, R. (2014). Decentralized learning for multiplayer multiarmed bandits. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, *60*(4), 2331–2345.

Koutsoupias, E., & Papadimitriou, C. (1999). Worst-case equilibria. In *Annual symposium on theoretical aspects of computer science* (pp. 404–413). Springer.

Lai, T. L., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, *6*(1), 4–22.

Magesh, A., & Veeravalli, V. V. (2021). Decentralized heterogeneous multi-player multi-armed bandits with non-zero rewards on collisions. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, *68*(4), 2622–2634.

Marden, J. R., Young, H. P., & Pao, L. Y. (2014). Achieving Pareto optimality through distributed learning. *SIAM Journal on Control and Optimization*, *52*(5), 2753–2770.

Milchtaich, I. (1996). Congestion games with player-specific payoff functions. *Games and Economic Behavior*, *13*(1), 111–124.

Monderer, D., & Shapley, L. S. (1996). Potential games. *Games and Economic Behavior*, *14*(1), 124–143.

Naparstek, O., & Leshem, A. (2013). Fully distributed optimal channel assignment for open spectrum access. *IEEE Transactions on Signal Processing*, *62*(2), 283–294.

Owen, G. (1968). *Game theory*. B, Saunders Co.

Pradelski, B. S., & Young, H. P. (2012). Learning efficient Nash equilibria in distributed systems. *Games and Economic Behavior*, *75*(2), 882–897.

Rosenski, J., Shamir, O., & Szlak, L. (2016). Multi-player bandits–a musical chairs approach. In *International conference on machine learning* (pp. 155–163).

Rosenthal, R. W. (1973). A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, *2*(1), 65–67.

Suri, S., Tóth, C. D., & Zhou, Y. (2004). Uncoordinated load balancing and congestion games in P2P systems. In *International workshop on peer-to-peer systems* (pp. 123–130). Springer.

Tang, F., Yang, L. T., Tang, C., Li, J., & Guo, M. (2018). A dynamical and load-balanced flow scheduling approach for big data centers in clouds. *IEEE Transactions on Cloud Computing*, *6*(4), 915–928.

Tibrewal, H., Patchala, S., Hanawal, M. K., & Darak, S. J. (2019). Multiplayer multi-armed bandits for optimal assignment in heterogeneous networks. arXiv preprint arXiv:1901.03868.

Wainwright, M. J. (2019). *Vol. 48, High-dimensional statistics: a non-asymptotic viewpoint*. Cambridge University Press.

Zafaruddin, S., Bistritz, I., Leshem, A., & Niyato, D. (2019). Distributed learning for channel allocation over a shared spectrum. *IEEE Journal on Selected Areas in Communications*, *37*(10), 2337–2349.

Amir Leshem (IEEE Fellow) received his B.Sc. (Cum Laude) in mathematics and physics, his M.Sc. (Cum Laude) in mathematics, and his Ph.D. in mathematics all from the Hebrew University, Jerusalem, Israel, in 1986, 1990 and 1998 respectively. From 1998 to 2000 he was with Faculty of Information Technology and Systems, Delft University of Technology, The Netherlands, as a postdoctoral researcher working on problems of signal processing for radio astronomy. From 2000 to 2003 he was the director of advanced technologies at Metalink Broadband where he was responsible for the research and development of new DSL and wireless MIMO modem technologies and served as a member of several international standard-setting groups such as ETSI TM06 and ITU-T SG15. From 2000 to 2002 he was also a visiting researcher at Delft University of Technology. In 2002 he joined Bar-Ilan University where he was one of the founders of the Faculty of Engineering and a full professor. He was head of the Signal Processing and Communications tracks, 2002–2014 and 2014–2017 respectively. In 2009 he spent his sabbatical at Delft University of Technology and Stanford University. Prof. Leshem was an associate editor of IEEE Trans. on Signal Processing 2008–2011, and the leading guest editor of several special issues of the IEEE Signal Processing Magazine and the IEEE Journal on Selected Topics in Signal Processing. From 2017 to 2021 he was associate editor for IEEE Trans. on Signal and information processing over networks. On January 2022 he was elevated to IEEE Fellow for contributions to multi-channel and multi-agent signal processing.

His main research interests include multi-agent learning over networks and multi-agent learning under limited communications, models for opinion dynamics and information propagation in social networks, wireless networks, applications of game theory to networks, signal, and information processing networks with applications to sensor and social networks, multichannel wireless and wireline communication, array and statistical signal, radio-astronomical imaging, set theory, logic and the foundations of mathematics.

Vikram Krishnamurthy (IEEE Fellow) received his B.E. degree from the University of Auckland in Electrical Engineering in 1988, and Ph.D. degree from the Australian National University in 1992 in mathematical systems theory. He is a professor in the School of Electrical \& Computer Engineering, Cornell University. From 2002–2016 he was a Professor and Canada Research Chair at the University of British Columbia, Canada.

His research interests include statistical signal processing and stochastic control in social networks and adaptive sensing.

He served as a Distinguished Lecturer for the IEEE Signal Processing Society and Editor-in-Chief of the IEEE Journal on Selected Topics in Signal Processing.

In 2013, he was awarded an Honorary Doctorate from KTH (Royal Institute of Technology), Sweden. He is author of the book Partially Observed Markov Decision Processes published by Cambridge University Press in 2016.

Tomer Boyarski received a B.Sc. in physics (2017) and M.Sc. in Electrical Engineering (2021) both cum laude from Bar-Ilan University, Israel.