Dynamic Games and Applications
https://doi.org/10.1007/s13235-025-00676-w

Online Multi-player Resource-Sharing Games with Bandit Feedback

Mevan Wijewardena[1] · Michael J. Neely[1]

Received: 11 October 2024 / Accepted: 15 September 2025

Abstract

This paper considers an online multi-player resource-sharing game with bandit feedback. Multiple players choose from a finite collection of resources in a time slotted system. In each time slot, each resource brings a random reward that is equally divided among the players who choose it. The reward vector is independent and identically distributed over the time slots. The statistics of the reward vector are unknown to the players. During each time slot, for each resource chosen by the first player, they receive as feedback the reward of the resource and the number of players who chose it, after the choice is made. We develop a novel Upper Confidence Bound (UCB) algorithm that learns the mean rewards using the feedback and maximizes the worst-case time-average expected reward of the first player. The algorithm gets within $\mathcal{O}(\log(T)/\sqrt{T})$ of optimality within T time slots. The simulations depict fast convergence of the learnt policy in comparison to the worst-case optimal policy.

Keywords Congestion games · Potential games · Fair reward allocation · Worst-case expected reward maximization

1 Introduction

In this paper, we consider the following game with $m \geq 2$ players numbered $1, 2, \dots, m$, and $n \geq 2$ resources numbered $1, 2, \cdots, n$. The game evolves in slotted time $t \in \{1, 2, \dots\}$. The vector $\boldsymbol{W}(t) \in \mathbb{R}^n$ denotes the random reward vector at time $t \in \{1, 2, \dots\}$. In particular, for each $i \in \{1, 2, \dots, n\}$ and each $t \in \{1, 2, \dots\}$, $W_i(t) \geq 0$ denotes the reward offered by resource *i* at time *t*. We assume that $\boldsymbol{W}(t)$ are i.i.d. with $\mathbb{E}\{\boldsymbol{W}(t)\} = \boldsymbol{E} = [E_1, E_2, \dots, E_n]$. The vector $\boldsymbol{E}$ is unknown to the players. During each time slot, each player selects *r* resources

✉ Mevan Wijewardena
mpathira@usc.edu

Michael J. Neely
mjneely@usc.edu

[1] Department of Electrical and Computer Engineering, University of Southern California, 3740 McClintock Ave, Los Angeles, California 90089, USA

Published online: 24 November 2025

without knowing the other player's selections (assume that $0 < r \leq n$), and without knowledge of $\boldsymbol{W}(t)$. During time slot *t*, for each $k \in \{1, 2, \dots, n\}$, each player selecting resource *k* receives a reward of $W_k(t)/S_k(t)$ from resource *k*, where $S_k(t)$ is the number of players choosing resource *k* during time slot *t*. For each $i \in \{1, 2, \dots, m\}$, let $\mathcal{A}_i(t)$ denote the set of resources chosen by player *i* during time slot *t*. During time slot *t*, after the selection of resources, player *i* receives $(W_k(t), S_k(t))$ for $k \in \mathcal{A}_i(t)$ as feedback.

The total reward received by player *i* during time slot *t* is $\sum_{k \in \mathcal{A}_i(t)} W_k(t)/S_k(t)$. The time-average expected reward of player *i* in a finite time horizon of *T* time slots is

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left\{ \sum_{k \in \mathcal{A}_i(t)} \frac{W_k(t)}{S_k(t)} \right\}.$$

The goal is to design policies to maximize the time-average expected reward of player 1. However, this is not possible since player 1 does not have control over the policies of the other players. Hence, we focus on maximizing the *worst-case* time-average expected reward of player 1, which we define in the sections below.

For $k \in \{1, 2, \dots, n\}$ and $t \in \{1, 2, \dots\}$, define $X_k(t) = \sum_{i=2}^{m} 1_{[k \in \mathcal{A}_i(t)]}$. Hence, $X_k(t)$ is the number of players (other than player 1) choosing resource *k* during time slot *t*. Also, we have $S_k(t) = 1_{[k \in \mathcal{A}_1(t)]} + X_k(t)$. For each *t*, it can be shown that $\boldsymbol{X}(t) \in \mathcal{J}$, where

$$\mathcal{J} = \left\{ \boldsymbol{x} \in \{0, 1, \dots, m-1\}^n \middle| \sum_{j=1}^{n} x_j = (m-1)r \right\}. \tag{1}$$

Additionally, for a given $t \in \{1, 2, \dots\}$, it can be easily shown that for any $\boldsymbol{x} \in \mathcal{J}$, there exists a way for players 2 to *m* to choose resources such that $\boldsymbol{X}(t) = \boldsymbol{x}$.

1.1 Time Average Expected Reward

Define $\mathcal{H}(t) = \{(\mathcal{A}_1(\tau), \{W_k(\tau), S_k(\tau); 1 \leq k \leq n, k \in \mathcal{A}_1(\tau)\}); 1 \leq \tau < t\}$, the history up to time *t*. Given $\mathcal{H}(t)$, the action of player 1 at time *t* is conditionally independent of the other player's actions at time *t*. Define the random vector $\boldsymbol{p}(t)$ with components $p_k(t) = \mathbb{E}\{1_{[k \in \mathcal{A}_1(t)]} | \mathcal{H}(t)\}$. Since $\sum_{k=1}^{n} 1_{[k \in \mathcal{A}_1(t)]} = r$, it can be shown that $\boldsymbol{p}(t) \in \Delta_{n,r}$, where $\Delta_{n,r}$ is the (*n*, *r*)-hypersimplex given by

$$\Delta_{n,r} = \left\{ \boldsymbol{p} \in \mathbb{R}_+^n : \sum_{i=1}^{n} p_i = r, p_i \in [0, 1] \ \forall i \in \{1, 2, \dots, n\} \right\}. \tag{2}$$

Also, notice that given $\boldsymbol{p} \in \Delta_{n,r}$, we can use the Madow's sampling technique (see for example [1]) to sample an action set $\mathcal{A} \subset \{1, 2, \dots, n\}$ such that, $|\mathcal{A}| = r$, and $p_k = \mathbb{E}\{1_{[k \in \mathcal{A}]}\}$ for each $k \in \{1, 2, \dots, n\}$.

Notice that we can write the time-average expected reward *R*(*T*) of player 1 in a finite time horizon of *T* time slots as

$$\begin{aligned} R(T) &= \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{n}\mathbb{E}\left\{\frac{W_k(t)1_{[k\in\mathcal{A}_1(t)]}}{1+X_k(t)}\right\} =_{(a)} \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{n}E_k\mathbb{E}\left\{\frac{1_{[k\in\mathcal{A}_1(t)]}}{1+X_k(t)}\right\} \\ &= \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{n}E_k\mathbb{E}\left\{\mathbb{E}\left\{\frac{1_{[k\in\mathcal{A}_1(t)]}}{1+X_k(t)}\middle|\mathcal{H}(t)\right\}\right\} \\ &=_{(b)} \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{n}E_k\mathbb{E}\left\{\mathbb{E}\left\{\frac{1}{1+X_k(t)}\middle|\mathcal{H}(t)\right\}\mathbb{E}\left\{1_{[k\in\mathcal{A}_1(t)]}|\mathcal{H}(t)\right\}\right\} \\ &= \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{n}E_k\mathbb{E}\left\{\mathbb{E}\left\{\frac{1}{1+X_k(t)}\middle|\mathcal{H}(t)\right\}p_k(t)\right\} \\ &=_{(c)} \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{n}E_k\mathbb{E}\left\{\frac{p_k(t)}{1+X_k(t)}\right\} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\{f(\boldsymbol{p}(t),\boldsymbol{X}(t))\}, \end{aligned} \tag{3}$$

where (a) follows since $\boldsymbol{W}(t)$ is independent of the actions of players at time t, (b) follows since $1_{[k\in\mathcal{A}_1(t)]}$ is independent of $X_k(t)$ conditioned on $\mathcal{H}(t)$, (c) follows since $\boldsymbol{p}(t)$ is an $\mathcal{H}(t)$-measurable random variable, and the function $f:\mathbb{R}_+^n\times\mathbb{Z}_+^n\to\mathbb{R}$ is defined as

$$f(\boldsymbol{p},\boldsymbol{x}) = \sum_{k=1}^{n}\frac{E_k p_k}{1+x_k}. \tag{4}$$

The time-average expected reward of player 1 is

$$R := \lim\inf_{T\to\infty} R(T). \tag{5}$$

1.2 Worst-Case Time Average Expected Reward

Notice that since player 1 does not have access to $\boldsymbol{X}(t)$ when taking action during time slot t, they cannot directly maximize R defined in (5). But notice that for fixed $\boldsymbol{p}\in\Delta_{n,r}$, the worst-case value of $f(\boldsymbol{p},\boldsymbol{x})$ is $f^{\text{worst}}(\boldsymbol{p})$, where

$$f^{\text{worst}}(\boldsymbol{p}) = \min_{\boldsymbol{x}\in\mathcal{J}} f(\boldsymbol{p},\boldsymbol{x}). \tag{6}$$

Combining with (3), the worst-case time-average expected reward in a finite time horizon of T time slots is given by

$$R^{\text{worst}}(T) = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\{f^{\text{worst}}(\boldsymbol{p}(t))\}. \tag{7}$$

Hence, the worst-case time-average expected reward of player 1 is

$$R^{\text{worst}} = \lim\inf_{T\to\infty} R^{\text{worst}}(T). \tag{8}$$

Instead of maximizing R, player 1 can take decisions to maximize R^{worst} without knowledge of the decisions of other players.

From (7) and (8), we have that the maximum possible value of R^{worst} is $f^{\text{worst},*}$, where

$$f^{\text{worst},*} = \max_{\boldsymbol{p}\in\Delta_{n,r}} \min_{\boldsymbol{x}\in\mathcal{J}} f(\boldsymbol{p},\boldsymbol{x}) = \max_{\boldsymbol{p}\in\Delta_{n,r}} f^{\text{worst}}(\boldsymbol{p}), \tag{9}$$

that is achieved by using the policy $\boldsymbol{p}(t) = \boldsymbol{p}^*$ in each time slot, where

$$\boldsymbol{p}^* \in \arg\max_{\boldsymbol{p}\in\Delta_{n,r}} f^{\text{worst}}(\boldsymbol{p}). \tag{10}$$

The f^{worst} function is unknown to player 1 because the function f defined in (4) is in terms of the unknown E_k values. Hence, we aim to design an algorithm that achieves a worst-case time-average expected reward close to $f^{\text{worst},*}$ using the bandit feedback.[1]

1.3 Related Work

The main challenge of applying online optimization techniques such as online gradient descent [2] to the above problem is due to the fact that we do not know the function f since we do not know $\boldsymbol{E}$. The problem shares certain similarities with the problems of multi-armed bandit learning (MAB) [3, 4], adversarial bandit learning [5], online-convex optimization [6], online-convex optimization with multi-point bandit feedback [7], and stochastic convex optimization [8].

Multi-armed bandit learning is extensively studied in the literature. The classical MAB problem consists of a fixed number of arms each with fixed mean reward. A player chooses an arm in each iteration of the game, without knowledge about the mean rewards, where after the choice is made the reward of the chosen arm is revealed to the player. The goal is to learn to choose the arm with the highest mean reward. An algorithm for the MAB problem has to explore all the arms in order to learn the best arm. But in doing so, the player also chooses arms with low mean reward, which affects the long term reward of the player. Upper confidence bound based algorithms, where the algorithm maintains an upper bound on the mean cost of each arm, are popular in the MAB literature [5, 9]. Our problem cannot be addressed using classical MAB approaches since the reward not only depends on the chosen resource, but also on the choices of other players. Another related problem is adversarial bandit learning. Unlike the worst-case approach, the adversarial bandit framework cannot be used to obtain utility guarantees for player 1 that are independent of the actions of the other players.

The framework of online optimization also shares similarities with our work since our goal is to design an online algorithm to minimize $f^{\text{worst}}(\boldsymbol{p})$. However, notice that f^{worst} depends on the unknown vector $\boldsymbol{E}$. We also do not have access to an unbiased estimate or an unbiased gradient estimate of the function f^{worst} due to its definition in (6). Hence, the work on online-convex optimization where partial information on the underlying reward

[1] One can relax the constraint of each user choosing exactly r resources by allowing each user to select at most r resources. Since the rewards are assumed to be nonnegative, this will not affect f^{worst}. A formal proof of this statement can also be found in our technical report [52].

functions are revealed, such as online-convex optimization with multi-point bandit feedback, and the approaches based on stochastic gradient descent are also not applicable. Our problem is more similar to the work of [10] on zero-sum matrix games with bandit feedback. However, the above work considers a two-player scenario where both players receive the actions and the rewards of themselves and the opponent as feedback.

Our game model has been studied for the offline non-stochastic case with full information on $\boldsymbol{E}$ under the more general framework of resource-sharing games [11], also known as congestion games. In these games, the *per-player reward* of a resource is a general function of the number of players selecting the resource. Also, an action for a player is a subset of the resources, where the allowed subsets make up the player's action space. Resource-sharing games have also been extended to various stochastic settings [12, 13]. Problems similar to our work have been studied in the context of adversarial resource-sharing games. The work of [14] considers an adversarial resource-sharing game where each player chooses a single resource from a collection of resources, after which an adversary chooses the resource chosen by the maximum number of players. Also, non-atomic congestion games with malicious players have been considered through the work of [15]. The above works assume that $\boldsymbol{E}$ is known to all the players.

We have simplified the general resource-sharing game model described above in two ways. First, we assume a fair-reward allocation model, where we have assumed the existence of a reward for each resource, which is divided equally between the players selecting it. Second, we have assumed simple action spaces for players by allowing each player to select an arbitrary subset of r resources. Resource-sharing games with special *per-player reward* definitions have been considered in the literature. One such notable case is when the *per-player* reward of a resource is nondecreasing in the number of players selecting the resource. These games are called cost-sharing games [16]. The particular case when the total cost of a resource is divided equally among the players choosing it is called *fair cost-sharing games*. In such a model, a player would prefer to select resources selected by many players. In the fair reward allocation model considered in our work, players have the opposite incentive to select resources selected by a small number of players.

One application of our model is multiple access control (MAC) in communication systems, where multiple users access communication channels, and the data rate of a channel is shared amongst the users who select it [17–19]. Here, a channel can be shared using Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA), where in TDMA, the channel is time-shared among the users [20, 21], whereas in FDMA, the channel is frequency-shared among the users [22]. In both cases, the total data rate supported by the channel can be considered the reward of the channel. Here, limiting the number of channels accessed by a single user in a given time slot is desirable. Additionally, the channel data rate should be shared among the users accessing the channel.

The worst-case expected reward is an important objective different from Nash-equilibrium [23, 24] and correlated equilibrium [25–27]. The problem of finding an approximate Nash equilibrium of a congestion game with bandit feedback has been considered [28]. However, implementing the algorithms by [28] requires cooperation among players. In contrast, the worst-case approach requires no cooperation among the players. Additionally, player 1 does not have to make assumptions about other players' strategies. Hence, understanding the worst-case expected reward is important even when the other players are not necessarily playing to hurt player 1. However, in practice, some players play just to hurt others. One particular example arises in military communications. Consider a multiple access communication system used in a military setting (for instance, consider the TDMA scheme considered in [21], which has a similar structure to our model). Here, some users may transmit to disrupt the communication capabilities of other users. Our formulation is applicable even when the other users form a coalition with the intention of reducing the data rate of a single user. Another motivation for the worst-case objective of this paper is to quantify the degree of punishment that can be inflicted on a particular user. This value is useful, for example, in repeated game algorithms that design punishment modes into the strategy space in order to discourage deviant behavior [27, 29].

1.4 Background on Resource-Sharing Games

The resource-sharing game was first studied by [11]. These games, also called congestion games, fall under the general category of potential games [30]. In potential games, the effect of any player changing policies is captured by the change of a global potential function. Various extensions to the classical resource sharing game introduced by [11] have been studied in the literature [31]. Some such extensions are stochastic resource-sharing games [12, 13], weighted resource-sharing games [32], games with player-dependent reward allocation [33], games with resources having preferences over players [34], and singleton games, where each player is only allowed to choose a single resource [35, 36].

Also similar to resource-sharing games are resource allocation games [37, 38]. In these games, a resource must be fairly divided among claimants claiming a certain portion. There is also work combining resource-sharing games with bandits and strategic experimentation. The work of [39] considers a two-player game where players continually choose between their private risky arm and a shared safe arm. Only one player can activate the safe arm at any given time, which guarantees a payoff. This congestion effect on the safe arm gives rise to strategic consideration among the players. These works are based on the model of multi-agent, multi-armed bandit problems introduced by [40]. Here, multiple players are faced with the same multi-armed bandit problem. In contrast to the classic single-agent setting, players can learn from other players' feedback, resulting in some players being able to free-ride on other players' experiments. This phenomena induces strategic experimentation.

Resource-sharing games have applications in multiple-access [17], network selection [41], network design [42], spectrum sharing [43], resource sharing in wireless networks [44], load balancing networks [45], radio access selection [46], service chains [47], and congestion control [48]

1.5 Contributions

We study the problem of maximizing the worst-case time average expected reward of online resource-sharing games with a fair-reward allocation model in the presence of bandit feedback on the mean rewards of the resources. We assume a model where in each time slot, each player is allowed to choose any r element subset of the n available resources, and the reward of a resource is shared among the users selecting it. We propose a novel algorithm combining the upper confidence bound technique with Madow's sampling technique and Euclidean projection onto the (n, r)-hypersimplex, to maximize the worst-case time average expected reward of player 1. In particular, in each time slot of the algorithm, we find $\boldsymbol{p}(t)$ in the (n, r)-hypersimplex, after which we sample the r resources for player 1 using Madow's sampling technique. The algorithm gets within $\mathcal{O}(\log(T)/\sqrt{T})$ of optimality in a finite time-horizon of T time slots. The parameters of the algorithm do not depend on T. Hence, the above guarantee can be achieved even if the time horizon T is unknown.

1.6 Notation

We use calligraphic letters to denote sets. Vectors and matrices are denoted in boldface characters. For integers n and m, we denote by $[n : m]$ the set of integers between n and m inclusive. Also, we use $\mathbb{N} = \{1, 2, 3, \dots\}$ to denote the set of positive integers and $\mathbb{N}_0 = \{0, 1, 2, \dots\}$ to denote the set of non-negative integers.

2 Worst-Case Expected Reward Maximization

First we state our assumptions.

A1 The collection $\{\boldsymbol{W}(t); 1 \leq t\}$ is independent and identically distributed and satisfies $W_i(t) \geq 0$ for all $t \in \mathbb{N}$ and $i \in [1 : n]$. Our formulation does not require the components of $\boldsymbol{W}(t)$ to be mutually independent for a particular $t \geq 1$.

A2 We have $W_k(t) = E_k + \eta_k(t)$ for all $1 \leq k \leq n$, where $\eta_k(t)$ for $t \geq 1$ and $k \in [1 : n]$ are zero-mean, 1-sub-Gaussian random variables.

Before moving on to the main results of the paper, we consider the problem of finding $f^{\text{worst},*}$ and $\boldsymbol{p}^*$ when $\boldsymbol{E}$ is known, where $f^{\text{worst},*}$ and $\boldsymbol{p}^*$ are defined in (9) and (10), respectively.

2.1 Finding $\mathbf{f^{worst,*}}$ with Known E

If $\boldsymbol{E}$ is known, given $\boldsymbol{p} \in \Delta_{n,r}$, the problem of finding $f^{\text{worst}}(\boldsymbol{p})$ has been well studied in the literature. In particular, we can find $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathcal{J}} f(\boldsymbol{p}, \boldsymbol{x})$. In Appendix C, we provide the algorithm for completeness. Hence, we can use standard min-max optimization techniques such as min-oracle algorithm [49] to find $f^{\text{worst},*}$ and $\boldsymbol{p}^*$. Also, since $\mathcal{J}$ is a finite set, and the function $f(\cdot, \boldsymbol{x})$ is concave for all $\boldsymbol{x} \in \mathcal{J}$, from the Danskin's theorem (see proposition 5.4.9.(b) of [50]), we can calculate a subgradient of f^{worst} at $\boldsymbol{p} \in \Delta_{n,r}$ as $\nabla_{\boldsymbol{p}} f^{\text{worst}}(\boldsymbol{p}) = \nabla_{\boldsymbol{p}} f(\boldsymbol{p}, \boldsymbol{x}^*)$ where $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathcal{J}} f(\boldsymbol{p}, \boldsymbol{x})$. Hence, we can also use standard subgradient descent with Euclidean projections onto $\Delta_{n,r}$ (see Algorithm 4 to project onto $\Delta_{n,r}$) to find $f^{\text{worst},*}$.

The work of [51] finds $f^{\text{worst},*}$ and $\boldsymbol{p}^*$ explicitly for the case $m = 2, r = 1$. We discuss the solution of this case in Sect. 2.1.1. In Sect. 2.1.2, we extend this to the case $m = 3, r = 1$. These explicit solutions provide a fast way to find $f^{\text{worst},*}$ and provide insight into the structure of optimal $\boldsymbol{p}^*$. For these two sections we will use the notation $\Delta_n = \Delta_{n,1}$.

2.1.1 Case $m = 2, r = 1$

In the following theorem, we restate the result of [51].

Theorem 1 *Consider the special case $m = 2, r = 1$ with $n \in \mathbb{N}$. Without loss of generality, assume that $\boldsymbol{E}$ satisfies $E_k \geq E_{k+1}$ for all $k \in [1 : n - 1]$. Define the sequence*

$(V_i; 1 \leq i \leq n)$ according to $V_i = \frac{i-\frac{1}{2}}{\sum_{k=1}^{i} \frac{1}{E_k}}$. Let $v = \arg\max_{1\leq i\leq n} V_i$, where arg max *returns the least index in the case of ties. Then, $\mathbf{p}^*$ can defined by*

$$p_k^* = \begin{cases} \frac{\frac{1}{E_k}}{\sum_{j=1}^{v} \frac{1}{E_j}} & if \quad 1 \leq k \leq v \\ 0 & otherwise. \end{cases} \tag{11}$$

Proof See [51]. □

Assume player 1 follows the policy $\boldsymbol{p}^*$ in Theorem 1. It can be shown that the worst-case for $\boldsymbol{p}^*$ occurs when player 2 always chooses resource 1. Consider the strategy profile where player 1 uses $\boldsymbol{p}^*$ and player 2 always chooses resource 1. It can be shown that player 2 cannot increase their reward by unilaterally deviating from the above profile. See the technical report [52] for the proof of this fact. Notice that this may not be a Nash equilibrium since $\boldsymbol{p}^*$ may not be the best response of player 1 to the strategy of player 2. However, this property incentivizes player 2 to use the above strategy, even if they do not care about hurting player 1. The above property is not true for general *m*, *r*. When $m > 2$, players [2 : *m*] can increase the congestion of resources with high mean rewards to reduce the expected reward of player 1. In such a scenario, a player in [2 : *m*] may increase their reward by switching to a resource with a less mean reward and less congestion. However, there are special cases in which the above property is true even when $m > 2$. One such example is discussed in the case $m = 3, r = 1$.

2.1.2 Case $m = 3, r = 1$

We first focus on solving the problem $\min_{\boldsymbol{x}\in\mathcal{J}} f(\boldsymbol{p}, \boldsymbol{x})$ for given $\boldsymbol{p} \in \Delta_n$.

Lemma 1 *Consider the special case $m = 3, r = 1$ with $n \in \mathbb{N}$ satisfying $n \geq 2$, and a fixed $\boldsymbol{p} \in \Delta_n$. Let $a = \arg\max_{1\leq i\leq n} E_i p_i$, and $b = \arg\max_{1\leq i\leq n, i\neq a} E_i p_i$, where we assume that* arg max *returns the least index in the case of ties. Define the two vectors $\boldsymbol{x}^1, \boldsymbol{x}^2 \in \mathcal{J}$, where*

$$x_k^1 = \begin{cases} 2 & if k = a, \\ 0 & otherwise, \end{cases} \quad and\, x_k^2 = \begin{cases} 1 & if k \in \{a, b\}, \\ 0 & otherwise. \end{cases} \tag{12}$$

Then $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathcal{J}} f(\boldsymbol{p}, \boldsymbol{x})$ can be given in two cases. 1) $E_a p_a \geq 3E_b p_b$: We have $\boldsymbol{x}^* = \boldsymbol{x}^1$. 2) $E_a p_a < 3E_b p_b$: We have $\boldsymbol{x}^* = \boldsymbol{x}^2$.

Proof Since $\boldsymbol{x}^* \in \mathcal{J}$, we know $\boldsymbol{x}^*$ has nonnegative components that sum to 2. If $\boldsymbol{x}^*$ has only one nonzero component at some index $k \in \{1, \ldots, n\}$, then $x_k^* = 2$ and *f* is minimized by choosing $k = a$, so assignment $\boldsymbol{x}^1$ holds. Else, $\boldsymbol{x}^*$ has exactly two nonzero components at indices $k, j \in [1 : n]$ $(k \neq j)$ and *f* is minimized by choosing $k = a$ and $j = b$, so assignment $\boldsymbol{x}^2$ holds. It remains to compare the two assignments.

Under $\boldsymbol{x}^1$: $f(\boldsymbol{p}, \boldsymbol{x}^1) = \frac{p_a E_a}{3} + p_b E_b + \sum_{k\notin\{a,b\}} p_k E_k = \sum_{k=1}^{n} p_k E_k - \frac{2 p_a E_a}{3}$.

Under $\boldsymbol{x}^2$: $f(\boldsymbol{p}, \boldsymbol{x}^2) = \frac{p_a E_a}{2} + \frac{p_b E_b}{2} + \sum_{k\notin\{a,b\}} p_k E_k = \sum_{k=1}^{n} p_k E_k - \frac{p_a E_a}{2} - \frac{p_b E_b}{2}$

Comparing the two cases, we have that for assignment $\boldsymbol{x}^1$, we require $E_a p_a \geq 3E_b p_b$ and for assignment $\boldsymbol{x}^2$, we require $E_a p_a < 3E_b p_b$. Hence, we are done. □

The following theorem introduces the solution of the case $m = 3, r = 1$.

Theorem 2 *Consider the special case $m = 3, r = 1$ with $n \in \mathbb{N}$ satisfying $n \geq 2$. Without loss of generality, assume that $\boldsymbol{E}$ satisfies $E_k \geq E_{k+1}$ for all $k \in [1 : n-1]$. Define the two sequences $(U_i; 1 \leq i \leq n)$ and $(V_i; 2 \leq i \leq n)$ according to $U_i = \frac{i}{\frac{3}{E_1} + \sum_{k=2}^{i} \frac{1}{E_k}}$ and $V_i = \frac{i-1}{\sum_{k=1}^{i} \frac{1}{E_k}}$. Let $u = \arg\max_{1 \leq i \leq n} U_i$, and $v = \arg\max_{2 \leq i \leq n} V_i$, where $\arg\max$ returns the least index in the case of ties. Then, $\mathbf{p}^*$ can be described under two cases.*

Case 1: *If $V_v > U_u$,*

$$p_k^* = \begin{cases} \frac{\frac{1}{E_k}}{\sum_{j=1}^{v} \frac{1}{E_j}} & if \quad 1 \leq k \leq v \\ 0 & otherwise. \end{cases} \tag{13}$$

Case 2: *If $U_u \geq V_v$,*

$$p_k^* = \begin{cases} \frac{\frac{3}{E_1}}{\frac{3}{E_1} + \sum_{j=2}^{u} \frac{1}{E_j}} & if\, k = 1 \\ \frac{\frac{1}{E_k}}{\frac{3}{E_1} + \sum_{j=2}^{u} \frac{1}{E_j}} & if\, 2 \leq k \leq u \\ 0 & otherwise. \end{cases} \tag{14}$$

Proof Given in Appendix B. □

It is interesting to note the variation of choice probabilities in both cases of the theorem. In case 1, player 1 first chooses a set of resources $[1 : v]$ (v resources with the highest mean rewards) to be chosen with nonzero probability. Then player 1 assigns probability p_k^* for $k \in [1 : v]$ such that the $p_k^* \propto 1/E_k$. This behavior can be explained as follows. First, player 1 never chooses resources with mean rewards below a certain threshold. Second, within the collection of resources with relatively high mean rewards, player 1 is tempted to choose resources with lower mean rewards with high probability since, in the worst case, opponents choose the rewards with the highest mean rewards.

In Case 2, a similar behavior can be observed. Player 1 first chooses a set of resources $[1 : u]$ to be chosen with nonzero probability. However, now player 1 assigns probability p_k^* for $k \in [1 : u]$ such that $p_1^* \propto 3/E_1$ and $p_k^* \propto 1/E_k$ for $k \in [2 : u]$. In particular, player 1 chooses the first resource with a higher probability. To see this clearly, consider the two scenarios in Fig. 1, where we consider two possibilities of $\boldsymbol{E}$ for $n = 10$ (Scenario 1 and Scenario 2). Figure 1-Left denotes the plot of $\boldsymbol{E}$ for the two scenarios. Although in the two scenarios, $\boldsymbol{E}$ is different only in E_1 by 0.1, Scenario 1 belongs to Case 1 of Theorem 2, whereas Scenario 2 belongs to Case 2. Figure 1-Right shows the higher choice probability of resource 1 in Scenario 2. Here, the mean reward of the first resource is high enough to give a high per-player reward even if many players select it.

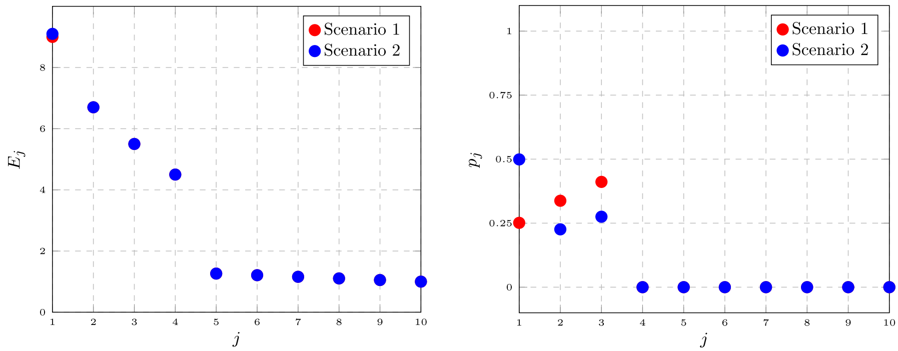

Fig. 1 Left: The mean rewards of the resources, Right: Probabilities of choosing the resources

Notice that in both scenarios, we have $1 \in \arg\max_{1\leq i\leq n} E_i p_i^*$, and $2 \in \arg\max_{1\leq i\leq n, i\neq 1} E_i p_i^*$. Also, since Scenario 1 belongs to Case 1 of Theorem 2, we have $p_1^* E_1 < 3p_2^* E_2$. Hence, from Lemma 1, we have that the worst-case for $\boldsymbol{p}^*$ occurs when player 2 always chooses resource 1 and player 3 always chooses resource 2 (or vice versa). Using a similar argument, one can establish that in Scenario 2, the worst-case for player 1 occurs when player 2 and player 3 always choose resource 1. Assuming player 1 plays the strategy $\boldsymbol{p}^*$, and the other two players play the strategies that give the worst-case to $\boldsymbol{p}^*$, the expected reward vector of the three players in Scenario 1 is (4.52, 7.87, 5.57). It turns out that in the above strategy profile, the strategies of players 2 and 3 are the best responses for the other two players. In fact this property holds more generally when $m = 3, r = 1$, and the solution comes from Case 1 of Theorem 2 as proved in [52]. However this is not true in Scenario 2, where the same vector is (4.54, 3.79, 3.79).

2.2 Bandit Algorithm

Now, we move on to the algorithm and analysis. Before introducing the algorithm, we begin with a few definitions and some preliminary results that are useful.

Our algorithm, provided in Algorithm 1 below, uses the first n time slots as an initial exploration phase that obtains at least one sample of the reward of each of the n resources. The main part of the algorithm starts in time slot $n + 1$.

For all $t \in \{n+1, n+2, \dots\}$ and $k \in [1:n]$ define $n_k(t)$ as the number of times player 1 chooses resource k before time slot t. Formally, $n_k(t) = \sum_{\tau=1}^{t-1} 1_{[k\in\mathcal{A}_1(\tau)]}$, where $\mathcal{A}_1(t)$ denotes the set of resources chosen by player 1 during time slot t. Notice that due to initial exploration phase of Algorithm 1, we have that $n_k(t) \geq 1$ for all $k \in [1:n]$ and $t \in \{n+1, n+2, \dots\}$. For each $t \in \{n+1, n+2, \dots\}$ and $k \in [1:n]$, define

$$\bar{E}_k(t) = \frac{1}{n_k(t)} \sum_{\tau=1}^{t-1} 1_{[k\in\mathcal{A}_1(\tau)]} W_k(\tau). \tag{15}$$

Fix $\delta_t \in (0, 1)$ for each $t \in \{n+1, n+2, \dots\}$ such that $\delta_t \geq \delta_{t+1}$ for all $t \in \{n+1, n+2, \dots\}$. For each $t \in \{n+1, n+2, \dots\}$ and $k \in [1:n]$, define

$$\tilde{E}_k(t) = \bar{E}_k(t) + \sqrt{\frac{2\log\left(\frac{n_k(t)(n_k(t)+1)}{\delta_t}\right)}{n_k(t)}}. \tag{16}$$

Also, define the functions $f_t : \mathbb{R}^n \times \mathbb{N}_0^n \to \mathbb{R}$ for $t \in \{n+1, n+2, \dots\}$ as

$$f_t(\boldsymbol{p}, \boldsymbol{x}) = \sum_{k=1}^{n} \frac{\tilde{E}_k(t) p_k}{1 + x_k}. \tag{17}$$

Before moving on to the main result, we introduce the following well-known lemma.

Lemma 2 *Given a sequence $\{X_t\}_{t=1}^{\infty}$ of independent zero-mean 1-sub Gaussian random variables, a positive integer-valued random variable G (possibly dependent on the sequence $\{X_t\}_{t=1}^{\infty}$) and $\epsilon \in (0, 1)$, we have*

$$\mathbb{P}\left\{\frac{1}{G}\sum_{i=1}^{G} X_i \le -\sqrt{\frac{2\log\left(\frac{G(G+1)}{\epsilon}\right)}{G}}\right\} \le \epsilon, \mathbb{P}\left\{\frac{1}{G}\sum_{i=1}^{G} X_i \ge \sqrt{\frac{2\log\left(\frac{G(G+1)}{\epsilon}\right)}{G}}\right\} \le \epsilon.$$

Proof This result is given as an exercise in the work of [5]. See [52] for a proof. □

Fix $t \in \{n+1, n+2, \dots\}$ and $k \in [1:n]$. For each $s \in [1:n_k(t)]$, define $\tilde{W}_k(s)$ as the reward obtained when the resource *k* is chosen for the *s*-th time by player 1. Hence, notice that $\bar{E}_k(t) = \frac{1}{n_k(t)} \sum_{s=1}^{n_k(t)} \tilde{W}_k(s)$. Notice that from assumption **A2**, $\{\tilde{W}_k(s) - E_k\}_{s=1}^{n_k(t)}$ is a collection of independent 1-sub Gaussian random variables. Applying Lemma 2 to $\{\tilde{W}_k(t) - E_k\}_{t=1}^{n_k(t)}$ with $G = n_k(t)$ and $\epsilon = \delta_t$, we have

$$\mathbb{P}\left\{\frac{1}{n_k(t)}\sum_{s=1}^{n_k(t)} (\tilde{W}_k(s) - E_k) \le -\sqrt{\frac{2\log\left(\frac{n_k(t)(n_k(t)+1)}{\delta_t}\right)}{n_k(t)}}\right\} \le \delta_t, \tag{18}$$

and

$$\mathbb{P}\left\{\frac{1}{n_k(t)}\sum_{s=1}^{n_k(t)} (\tilde{W}_k(s) - E_k) \ge \sqrt{\frac{2\log\left(\frac{n_k(t)(n_k(t)+1)}{\delta_t}\right)}{n_k(t)}}\right\} \le \delta_t. \tag{19}$$

The above two inequalities translate to

$$\mathbb{P}\left\{E_k \ge \tilde{E}_k(t)\right\} \le \delta_t, \tag{20}$$

and

$$\mathbb{P}\left\{E_k \le \tilde{E}_k(t) - 2\sqrt{\frac{2\log\left(\frac{n_k(t)(n_k(t)+1)}{\delta_t}\right)}{n_k(t)}}\right\} \le \delta_t, \tag{21}$$

for all $t \in \{n+1, n+2, \dots\}$ and $k \in [1:n]$, where $\tilde{E}_k(t)$ is defined in (16).

Now consider the collection $\{G_{n+1}, G_{n+2}, \dots\}$ of events that shall be called “good" events: For $t \in \{n+1, n+2, \dots\}$ the “good" event G_t is defined by the inequalities

$$E_k < \tilde{E}_k(t), \tag{22}$$

and

$$E_k > \tilde{E}_k(t) - 2\sqrt{\frac{2\log\left(\frac{n_k(t)(n_k(t)+1)}{\delta_t}\right)}{n_k(t)}} \tag{23}$$

for $k \in [1:n]$. Specifically, G_t is defined as the event that (22) and (23) hold for all $k \in [1:n]$. Combining (20) and (21) with the union bound, we have that

$$\mathbb{P}\{G_t^c\} \le 2n\delta_t. \tag{24}$$

Recall that

$$\boldsymbol{p}^* \in \arg\max_{\boldsymbol{p}\in\Delta_{n,r}} f^{\text{worst}}(\boldsymbol{p}), \tag{25}$$

where the function f^{worst} is defined in (6). Let

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathcal{J}} f(\boldsymbol{p}^*, \boldsymbol{x}), \tag{26}$$

where the function f is defined in (4). Hence, we have that

$$f^{\text{worst},*} = f(\boldsymbol{p}^*, \boldsymbol{x}^*), \tag{27}$$

where $f^{\text{worst},*}$ is defined in (9). Before moving on to the Algorithm and the main theorem, we first prove the following lemma.

Lemma 3 *Fix $t \in \{n+1, n+2, \dots\}$. Assume that the “good" event G_t is true. Then we have that*

(a) *$f_t(\boldsymbol{p}^*, \boldsymbol{x}) \ge f(\boldsymbol{p}^*, \boldsymbol{x}^*)$ for every $\boldsymbol{x} \in \mathcal{J}$, where f_t is defined in (17), $\mathbf{p}^*$ is defined in (25) and $\boldsymbol{x}^*$ is defined in (26).*

(b) *Define*

$$D_t = C + 2\sqrt{2\log\left(\frac{t(t+1)}{\delta_t}\right)}, \tag{28}$$

where

$$C = \max_{k\in[1:n]} E_k. \tag{29}$$

We have that $\|\nabla_{\boldsymbol{p}} f_t(\boldsymbol{p},\boldsymbol{x})\|^2 \le nD_t^2$ *for every* $\boldsymbol{p} \in \Delta_{n,r}$ *and* $\boldsymbol{x} \in \mathcal{J}$.

Proof We prove the two parts separately.

(a) We have that

$$f_t(\boldsymbol{p}^*,\boldsymbol{x}) = \sum_{k=1}^{n} \frac{\tilde{E}_k(t)p_k^*}{1+x_k} \ge_{(a)} \sum_{k=1}^{n} \frac{E_k p_k^*}{1+x_k} = f(\boldsymbol{p}^*,\boldsymbol{x}) \ge f(\boldsymbol{p}^*,\boldsymbol{x}^*), \tag{30}$$

where (a) follows since we are in the “good" event G_t (so (22) holds) and the last inequality follows from the definition of $\boldsymbol{x}^*$ in (26).

(b) First, notice that when we are in the event G_t, we have from (23) that,

$$\tilde{E}_k(t) < E_k + 2\sqrt{\frac{2\log\left(\frac{n_k(t)(n_k(t)+1)}{\delta_t}\right)}{n_k(t)}} \le_{(a)} C + 2\sqrt{2\log\left(\frac{t(t+1)}{\delta_t}\right)} = D_t, \tag{31}$$

for all $k \in [1:n]$, where (a) follows since $E_k \le C$ by definition of C in (29), and $1 \le n_k(t) \le t$ for $t \in \{n+1, n+2, \dots\}$ by definition of $n_k(t)$. Hence,

$$\|\nabla_{\boldsymbol{p}} f_t(\boldsymbol{p},\boldsymbol{x})\|^2 = \sum_{k=1}^{n} \frac{\tilde{E}_k^2(t)}{(1+x_k)^2} \le \sum_{k=1}^{n} \tilde{E}_k^2(t) \le nD_t^2. \tag{32}$$

□

We summarize our approach in Algorithm 1. The algorithm relies on three key steps.

First, we assume that given $\boldsymbol{p} \in \Delta_{n,r}$, we can sample a set $\mathcal{A} \subset [1:n]$ such that $|\mathcal{A}| = r$, and $\mathbb{E}\{1_{k\in\mathcal{A}}\} = p_k$ for all $k \in [1:n]$. This can be solved using the Madow’s sampling technique ([1]). In Appendix A, we provide the algorithm for completeness. The correctness of the algorithm is established in [1].

Second, we assume we have an oracle that can compute a solution $\boldsymbol{y} \in \arg\min_{\boldsymbol{x}\in\mathcal{J}} \sum_{k=1}^{n} \frac{F_k p_k}{1+x_k}$, where $F_i \ge 0$ for all $i \in [1:n]$, and $\boldsymbol{p} = [p_1, p_2, \dots, p_n] \in \Delta_{n,r}$. This problem is nonconvex due to the fact that $\mathcal{J}$ is a discrete set. Nevertheless, the problems of above type can be solved explicitly (we describe a simple method in Appendix C).

Finally, we assume that given $\boldsymbol{x} \in \mathbb{R}_+^n$, we can find the projection $\Pi_{\Delta_{n,r}}(\boldsymbol{x})$ of $\boldsymbol{x}$ onto $\Delta_{n,r}$. An algorithm for this task is given in Appendix D along with the analysis.

For our algorithm, we also require step size parameters β_t for $t \in \{n+1, n+2, \dots\}$ satisfying $\beta_t \geq \beta_{t+1}$ for all $t \in \{n+1, n+2, \dots\}$.

```
for each time slot $t \in [1:n]$ do
    Set $\mathcal{A}_1(t) \subset [1:n]$ arbitrarily satisfying $|\mathcal{A}_1(t)| = r$ and $t \in \mathcal{A}_1(t)$.
    Receive feedback $\{W_k(t); 1 \leq k \leq n, k \in \mathcal{A}_1(t)\}$.
end
Initialize $\boldsymbol{p}(n+1) \in \Delta_{n,r}$.
for each time slot $t \in \{n+1, n+2, \dots,\}$ do
    Sample an action set $\mathcal{A}_1(t) \subset [1:n]$ using the Madow's sampling
      technique such that $|\mathcal{A}_1(t)| = r$, and $p_k(t) = \mathbb{E}\{\mathbb{1}_{[k \in \mathcal{A}_1(t)]} | \boldsymbol{p}(t)\}$ for each
      $k \in [1:n]$. In particular, given $\boldsymbol{p}(t)$, we sample the above action set
      independent of the past $\mathcal{H}(t)$ (see Appendix A for the implementation).
    Receive feedback $\{W_k(t); 1 \leq k \leq n, k \in \mathcal{A}_1(t)\}$.
    Find $\boldsymbol{x}(t)$ by solving,

          $\boldsymbol{x}(t) \in \arg\min_{\boldsymbol{x} \in \mathcal{J}} f_t(\boldsymbol{p}(t), \boldsymbol{x})$

      using Algorithm 3, where $f_t$ is defined in (17).
    Obtain $\boldsymbol{p}(t+1)$ by using,

          $\boldsymbol{p}(t+1) = \Pi_{\Delta_{n,r}} (\boldsymbol{p}(t) + \beta_t \nabla_{\boldsymbol{p}} f_t(\boldsymbol{p}(t), \boldsymbol{x}(t)))$ ,

      where $\Pi_{\Delta_{n,r}}(\boldsymbol{y})$ denotes the projection of $\boldsymbol{y}$ onto $\Delta_{n,r}$ (See Appendix 4
      for an algorithm) and $\beta_t$ is the step size parameter.
end
```

Algorithm 1 UCB based algorithm for worst-case maximization

2.3 Analysis of the Algorithm

In this section, we focus on establishing the performance of Algorithm 1.

Theorem 3 *Fix $T \in \{n+2, n+3, \dots\}$.*

(a) *Running the UCB based worst-case maximization algorithm in Algorithm 1 for T time slots with $\beta_t > 0$ such that $\beta_t \geq \beta_{t+1}$ and $\delta_t \in (0,1)$ such that $\delta_t \geq \delta_{t+1}$ for all $t \in \{n+1, n+2, \dots\}$ yields*

$$f^{\text{worst},*} - R^{\text{worst}}(T) \leq \frac{n}{2\beta_T T} + \frac{nrC}{T} + \frac{nD_T^2 \sum_{t=n+1}^{T} \beta_t}{2T} + 4\sqrt{\frac{2nr \log\left(\frac{T(T+1)}{\delta_T}\right)}{T}} + \frac{1}{T} \sum_{t=n+1}^{T} \left(2rC + \frac{n}{\beta_t}\right) n\delta_t,$$

where $R^{\text{worst}}(T)$ is the time-average worst case expected reward achieved by the algorithm (See (7)), C is defined in (29), and D_T is defined in (28). Notice that the algorithm does not require the knowledge of T. Hence, the algorithm can be implemented *in a setting where the time horizon is unknown.*

(b) *Running Algorithm 1 for T time slots with $\delta_t = \Theta(1/t)$ and $\beta_t = \Theta(1/\sqrt{t})$ for all $t \in \{n+1, n+2, \dots\}$, we have that $f^{\text{worst},*} - R^{\text{worst}}(T) \leq \Theta\left(\log(T)/\sqrt{T}\right)$.*

Proof We will first prove part-(a).

(a) Fix any $t \in \{n+1, \dots, T\}$ and assume the "good" event G_t holds. Lemma 3 implies $\|\nabla_{\boldsymbol{p}} f_t(\boldsymbol{p}(t), \boldsymbol{x}(t))\|^2 \leq nD_t^2 \leq nD_T^2$, where the first inequality follows from Lemma 3-(b) and the second inequality follows since $D_t \leq D_T$ for all $t \in \{n+1, \dots, T\}$ (see the definition of D_t in (28) and use the fact that $\delta_t \geq \delta_T$ for all $t \in \{n+1, \dots, T\}$). Also, we have

$$f_t(\boldsymbol{p}^*, \boldsymbol{x}(t)) \geq f(\boldsymbol{p}^*, \boldsymbol{x}^*) = f^{\text{worst},*}, \tag{33}$$

where $\boldsymbol{x}(t)$ is defined in line 9 of Algorithm 1, f_t is defined in (17), the first inequality follows from Lemma 3-(a) and the last equality follows from (27). Define

$$\tilde{\boldsymbol{x}}(t) \in \arg\min_{\boldsymbol{x} \in \mathcal{J}} f(\boldsymbol{p}(t), \boldsymbol{x}).$$

Thus

$$f(\boldsymbol{p}(t), \tilde{\boldsymbol{x}}(t)) = f^{\text{worst}}(\boldsymbol{p}(t)) \tag{34}$$

by the definition of f^{worst} in (6). Due to the definition of $\boldsymbol{x}(t)$ in line 9 of Algorithm 1, we have

$$f_t(\boldsymbol{p}(t), \boldsymbol{x}(t)) \leq f_t(\boldsymbol{p}(t), \tilde{\boldsymbol{x}}(t)). \tag{35}$$

Also, notice that

$$\begin{aligned} f_t(\boldsymbol{p}(t), \tilde{\boldsymbol{x}}(t)) &=_{(a)} \sum_{k=1}^{n} \frac{\tilde{E}_k(t) p_k(t)}{1 + \tilde{x}_k(t)} = \sum_{k=1}^{n} \frac{E_k p_k(t)}{1 + \tilde{x}_k(t)} + \sum_{k=1}^{n} \frac{[\tilde{E}_k(t) - E_k] p_k(t)}{1 + \tilde{x}_k(t)} \\ &=_{(b)} f^{\text{worst}}(\boldsymbol{p}(t)) + \sum_{k=1}^{n} \frac{[\tilde{E}_k(t) - E_k] p_k(t)}{1 + \tilde{x}_k(t)} \\ &\leq_{(c)} f^{\text{worst}}(\boldsymbol{p}(t)) + \sum_{k=1}^{n} \left(\frac{2p_k(t)}{1 + \tilde{x}_k(t)} \sqrt{\frac{2\log\left(\frac{n_k(t)(n_k(t)+1)}{\delta_t}\right)}{n_k(t)}} \right) \\ &\leq_{(d)} f^{\text{worst}}(\boldsymbol{p}(t)) + 2\sum_{k=1}^{n} \left(p_k(t) \sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_k(t)}} \right) \end{aligned} \tag{36}$$

where (a) follows from the definition of f_t in (17); (b) follows from (34); (c) follows since we assume the "good" event G_t holds (hence, the inequality (23) is true); (d) follows since

$n_k(t) \leq T$, $\delta_t \geq \delta_T$ for all $t \in \{n+1, \ldots, T\}$, and $\tilde{x}_k(t) \geq 0$ for all $k \in [1:n]$. Since $\boldsymbol{p}(t+1)$ is defined in line 10 of Algorithm 1 as the projection of $\boldsymbol{p}(t) + \beta_t \nabla_{\boldsymbol{p}} f_t(\boldsymbol{p}(t), \boldsymbol{x}(t))$ onto the convex set $\Delta_{n,r}$, we have that,

$$\begin{aligned}
&\|\boldsymbol{p}(t+1) - \boldsymbol{p}^*\|^2 \leq_{(a)} \|\boldsymbol{p}(t) + \beta_t \nabla_{\boldsymbol{p}} f_t(\boldsymbol{p}(t), \boldsymbol{x}(t)) - \boldsymbol{p}^*\|^2 \\
&\leq \|\boldsymbol{p}(t) - \boldsymbol{p}^*\|^2 + \beta_t^2 \|\nabla_{\boldsymbol{p}} f_t(\boldsymbol{p}(t), \boldsymbol{x}(t))\|^2 - 2\beta_t (\boldsymbol{p}^* - \boldsymbol{p}(t))^\top \nabla_{\boldsymbol{p}} f_t(\boldsymbol{p}(t), \boldsymbol{x}(t)) \\
&=_{(b)} \|\boldsymbol{p}(t) - \boldsymbol{p}^*\|^2 + \beta_t^2 \|\nabla_{\boldsymbol{p}} f_t(\boldsymbol{p}(t), \boldsymbol{x}(t))\|^2 - 2\beta_t (f_t(\boldsymbol{p}^*, \boldsymbol{x}(t)) - f_t(\boldsymbol{p}(t), \boldsymbol{x}(t))) \\
&\leq_{(c)} \|\boldsymbol{p}(t) - \boldsymbol{p}^*\|^2 + n\beta_t^2 D_T^2 - 2\beta_t f^{\text{worst},*} + 2\beta_t f_t(\boldsymbol{p}(t), \tilde{\boldsymbol{x}}(t))) \\
&\leq_{(d)} \|\boldsymbol{p}(t) - \boldsymbol{p}^*\|^2 + n\beta_t^2 D_T^2 + 4\beta_t \sum_{k=1}^{n} \left\{ p_k(t) \sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_k(t)}} \right\} \\
&\quad - 2\beta_t f^{\text{worst},*} + 2\beta_t f^{\text{worst}}(\boldsymbol{p}(t))
\end{aligned} \tag{37}$$

where (a) follows since projection onto the convex set $\Delta_{n,r}$ reduces the distance to any point in the set, (b) follows from the subgradient equality for the linear function $f_t(\cdot, \boldsymbol{x}(t))$, (c) follows from (33) and (35), and (d) follows from (36).

Hence, we have that for all $t \in \{n+1, \ldots, T\}$, given that the “good" event G_t is true

$$\begin{aligned}
&2f^{\text{worst},*} - 2f^{\text{worst}}(\boldsymbol{p}(t)) - \frac{1}{\beta_t}\|\boldsymbol{p}(t) - \boldsymbol{p}^*\|^2 + \frac{1}{\beta_t}\|\boldsymbol{p}(t+1) - \boldsymbol{p}^*\|^2 \\
&\leq n\beta_t D_T^2 + 4\sum_{k=1}^{n} \left\{ p_k(t) \sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_k(t)}} \right\}
\end{aligned} \tag{38}$$

Notice that

$$\begin{aligned}
p_k(t) &=_{(a)} \mathbb{E}\{1_{[k \in \mathcal{A}_1(t)]} | \boldsymbol{p}(t)\} =_{(b)} \mathbb{E}\{1_{[k \in \mathcal{A}_1(t)]} | \boldsymbol{p}(t), \mathcal{H}(t)\} \\
&=_{(c)} \mathbb{E}\{1_{[k \in \mathcal{A}_1(t)]} | \mathcal{H}(t)\},
\end{aligned} \tag{39}$$

where (a) follows due to the sampling of the set $\mathcal{A}_1(t)$ in line 7 of Algorithm 1, (b) follows because the action set $\mathcal{A}_1(t)$ is sampled independent of the history $\mathcal{H}(t)$ given $\boldsymbol{p}(t)$ (see line 7 of Algorithm 1), and (c) follows since $\boldsymbol{p}(t)$ is $\mathcal{H}(t)$-measurable.

Now we take the expectation (Conditioned on the event G_t) of both sides of (38) which gives,

$$\begin{aligned}
&\mathbb{E}\left\{2f^{\text{worst},*}-2f^{\text{worst}}(\boldsymbol{p}(t))-\frac{1}{\beta_t}\|\boldsymbol{p}(t)-\boldsymbol{p}^*\|^2+\frac{1}{\beta_t}\|\boldsymbol{p}(t+1)-\boldsymbol{p}^*\|^2\middle|G_t\right\}\\
&\le n\beta_t D_T^2+4\mathbb{E}\left\{\sum_{k=1}^{n}p_k(t)\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_k(t)}}\middle|G_t\right\}\\
&\le n\beta_t D_T^2+\frac{4}{\mathbb{P}\{G_t\}}\mathbb{E}\left\{\sum_{k=1}^{n}p_k(t)\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_k(t)}}\right\}\\
&=_{(a)}\ n\beta_t D_T^2+\frac{4}{\mathbb{P}\{G_t\}}\mathbb{E}\left\{\sum_{k=1}^{n}\mathbb{E}\{1_{[k\in\mathcal{A}_1(t)]}|\mathcal{H}(t)\}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_k(t)}}\right\}\\
&=_{(b)}\ n\beta_t D_T^2+\frac{4}{\mathbb{P}\{G_t\}}\mathbb{E}\left\{\mathbb{E}\left\{\sum_{k=1}^{n}1_{[k\in\mathcal{A}_1(t)]}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_k(t)}}\middle|\mathcal{H}(t)\right\}\right\}\\
&=n\beta_t D_T^2+\frac{4}{\mathbb{P}\{G_t\}}\mathbb{E}\left\{\mathbb{E}\left\{\sum_{j:j\in\mathcal{A}_1(t)}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_j(t)}}\middle|\mathcal{H}(t)\right\}\right\}\\
&=n\beta_t D_T^2+\frac{4}{\mathbb{P}\{G_t\}}\mathbb{E}\left\{\sum_{j:j\in\mathcal{A}_1(t)}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_j(t)}}\right\}
\end{aligned}\tag{40}$$

where (a) follows from (39) and (b) follows since $n_k(t)$ is $\mathcal{H}(t)$-measurable. Hence, we have that for $t\in\{n+1,\dots,T\}$

$$\begin{aligned}
\mathbb{E}\left\{2f^{\text{worst},*}-2f^{\text{worst}}(\boldsymbol{p}(t))|G_t\right\}&\le\frac{\mathbb{E}\{\|\boldsymbol{p}(t)-\boldsymbol{p}^*\|^2|G_t\}}{\beta_t}-\frac{\mathbb{E}\{\|\boldsymbol{p}(t+1)-\boldsymbol{p}^*\|^2|G_t\}}{\beta_t}\\
&+n\beta_t D_T^2+\frac{4}{\mathbb{P}\{G_t\}}\mathbb{E}\left\{\sum_{j:j\in\mathcal{A}_1(t)}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_j(t)}}\right\}.
\end{aligned}\tag{41}$$

Now, notice that

$$\begin{aligned}
&\mathbb{E}\{\|\boldsymbol{p}(t+1)-\boldsymbol{p}^*\|^2|G_t\}\mathbb{P}\{G_t\}=\mathbb{E}\{\|\boldsymbol{p}(t+1)-\boldsymbol{p}^*\|^2\}\\
&\quad-\mathbb{E}\{\|\boldsymbol{p}(t+1)-\boldsymbol{p}^*\|^2|G_t^c\}\mathbb{P}\{G_t^c\}\ge\mathbb{E}\{\|\boldsymbol{p}(t+1)-\boldsymbol{p}^*\|^2\}-n\mathbb{P}\{G_t^c\},
\end{aligned}\tag{42}$$

where the last inequality follows from $\|\boldsymbol{p}(t+1)-\boldsymbol{p}^*\|^2\le n$ (since $\boldsymbol{p}(t+1),\boldsymbol{p}^*\in\Delta_{n,r}$). Next, notice that

$$f^{\text{worst},*} = \sum_{k=1}^{n} \frac{p_k^* E_k}{1 + x_k^*} \le \sum_{k=1}^{n} p_k^* C = rC, \tag{43}$$

where the first equality follows from (27), the inequality follows from the definition of C in (29) and the fact that $x_k^* \ge 0$ for all $k \in [1:n]$, and the last equality follows since $\boldsymbol{p}^* \in \Delta_{n,r}$ (see (25)). Hence,

$$\mathbb{E}\left\{2f^{\text{worst},*} - 2f^{\text{worst}}(\boldsymbol{p}(t))|G_t^c\right\} \le 2f^{\text{worst},*} \le 2rC, \tag{44}$$

where the last inequality follows from (43). Notice that,

$$\begin{aligned}
&\mathbb{E}\{2f^{\text{worst},*} - 2f^{\text{worst}}(\boldsymbol{p}(t))\} \\
&= \mathbb{E}\{2f^{\text{worst},*} - 2f^{\text{worst}}(\boldsymbol{p}(t))|G_t\}\mathbb{P}\{G_t\} + \mathbb{E}\{2f^{\text{worst},*} - 2f^{\text{worst}}(\boldsymbol{p}(t))|G_t^c\}\mathbb{P}\{G_t^c\} \\
&\le_{(a)} \frac{1}{\beta_t}\mathbb{E}\{\|\boldsymbol{p}(t) - \boldsymbol{p}^*\|^2|G_t\}\mathbb{P}\{G_t\} - \frac{1}{\beta_t}\mathbb{E}\{\|\boldsymbol{p}(t+1) - \boldsymbol{p}^*\|^2|G_t\}\mathbb{P}\{G_t\} + n\beta_t D_T^2\mathbb{P}\{G_t\} \\
&\quad + 4\mathbb{E}\left\{\sum_{j:j\in\mathcal{A}_1(t)} \sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_j(t)}}\right\} + 2rC\mathbb{P}\{G_t^c\} \\
&\le_{(b)} \frac{1}{\beta_t}\mathbb{E}\{\|\boldsymbol{p}(t) - \boldsymbol{p}^*\|^2\} - \frac{1}{\beta_t}\mathbb{E}\{\|\boldsymbol{p}(t+1) - \boldsymbol{p}^*\|^2\} + \frac{n}{\beta_t}\mathbb{P}\{G_t^c\} + n\beta_t D_T^2 \\
&\quad + 4\mathbb{E}\left\{\sum_{j:j\in\mathcal{A}_1(t)} \sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_j(t)}}\right\} + 2rC\mathbb{P}\{G_t^c\} \\
&\le_{(c)} \frac{1}{\beta_t}\mathbb{E}\{\|\boldsymbol{p}(t) - \boldsymbol{p}^*\|^2\} - \frac{1}{\beta_t}\mathbb{E}\{\|\boldsymbol{p}(t+1) - \boldsymbol{p}^*\|^2\} + n\beta_t D_T^2 \\
&\quad + 4\mathbb{E}\left\{\sum_{j:j\in\mathcal{A}_1(t)} \sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_j(t)}}\right\} + 2\left(2rC + \frac{n}{\beta_t}\right)n\delta_t,
\end{aligned} \tag{45}$$

where (a) follows from (41) and (44), (b) follows since $\mathbb{E}\{X|Y\}\mathbb{P}\{Y\} \le \mathbb{E}\{X\}$ for a positive valued random variable X and (42), and (c) follows from (24). Now, we sum (45) for $t \in \{n+1, \ldots, T\}$ to get

$$\begin{aligned}
&\mathbb{E}\left\{2(T-n)f^{\text{worst},*}-2\sum_{t=n+1}^{T} f^{\text{worst}}(\boldsymbol{p}(t))\right\}\\
&\le \frac{\mathbb{E}\{\|\boldsymbol{p}(n+1)-\boldsymbol{p}^*\|^2\}}{\beta_{n+1}}+\sum_{t=n+2}^{T}\left[\frac{1}{\beta_t}-\frac{1}{\beta_{t-1}}\right]\mathbb{E}\{\|\boldsymbol{p}(t)-\boldsymbol{p}^*\|^2\}-\frac{\mathbb{E}\{\|\boldsymbol{p}(T+1)-\boldsymbol{p}^*\|^2\}}{\beta_{T+1}}\\
&\quad+nD_T^2\sum_{t=n+1}^{T}\beta_t+4\sum_{t=n+1}^{T}\mathbb{E}\left\{\sum_{j:j\in\mathcal{A}_1(t)}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_j(t)}}\right\}+\sum_{t=n+1}^{T}2\left(2rC+\frac{n}{\beta_t}\right)n\delta_t\\
&\le_{(a)}\frac{n}{\beta_{n+1}}+\sum_{t=n+2}^{T}n\left[\frac{1}{\beta_t}-\frac{1}{\beta_{t-1}}\right]+nD_T^2\sum_{t=n+1}^{T}\beta_t\\
&\quad+4\sum_{t=n+1}^{T}\mathbb{E}\left\{\sum_{j:j\in\mathcal{A}_1(t)}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_j(t)}}\right\}+\sum_{t=n+1}^{T}2\left(2rC+\frac{n}{\beta_t}\right)n\delta_t\\
&=\frac{n}{\beta_T}+nD_T^2\sum_{t=n+1}^{T}\beta_t+4\sum_{t=n+1}^{T}\mathbb{E}\left\{\sum_{j:j\in\mathcal{A}_1(t)}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_j(t)}}\right\}\\
&\quad+\sum_{t=n+1}^{T}2\left(2rC+\frac{n}{\beta_t}\right)n\delta_t
\end{aligned}\tag{46}$$

where (a) follows since $1/\beta_t-1/\beta_{t-1}\ge 0$ for all $t\in\{n+2,\dots,T\}$ and $\|\boldsymbol{p}(t)-\boldsymbol{p}^*\|^2\le n$ for all $t\in\{n+1,\dots,T\}$ (since $\boldsymbol{p}(t),\boldsymbol{p}^*\in\Delta_{n,r}$). Now, notice that

$$\begin{aligned}
&\sum_{t=n+1}^{T}\mathbb{E}\left\{\sum_{j:j\in\mathcal{A}_1(t)}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_j(t)}}\right\}=\mathbb{E}\left\{\sum_{k=1}^{n}\sum_{\substack{t=n+1\\ k:k\in\mathcal{A}_1(t)}}^{T}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{n_k(t)}}\right\}\\
&=\mathbb{E}\left\{\sum_{k=1}^{n}\sum_{j=n_k(n+1)}^{n_k(T)}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{j}}\right\}\le\mathbb{E}\left\{\sum_{k=1}^{n}\sum_{j=1}^{n_k(T)}\sqrt{\frac{2\log\left(\frac{T(T+1)}{\delta_T}\right)}{j}}\right\}\\
&\le_{(a)}2\mathbb{E}\left\{\sum_{k=1}^{n}\sqrt{2n_k(T)\log\left(\frac{T(T+1)}{\delta_T}\right)}\right\}\le_{(b)}2\sqrt{2n\log\left(\frac{T(T+1)}{\delta_T}\right)}\sqrt{\sum_{k=1}^{n}n_k(T)}\\
&\le_{(c)}2\sqrt{2nrT\log\left(\frac{T(T+1)}{\delta_T}\right)},
\end{aligned}$$

where (a) follows from $\sum_{j=1}^{l}\sqrt{j}^{-1}\le 2\sqrt{l}$, (b) follows since $\sum_{k=1}^{n}\sqrt{n_k(T)}\le\sqrt{n\sum_{k=1}^{n}n_k(T)}$, and (c) follows since $\sum_{k=1}^{n}n_k(T)=r(T-1)\le rT$. Substituting above in (46), we have that

$$\begin{aligned}&\mathbb{E}\left\{2(T-n)f^{\text{worst},*}-2\sum_{t=n+1}^{T}f^{\text{worst}}(\boldsymbol{p}(t))\right\}\\&\le\frac{n}{\beta_T}+nD_T^2\sum_{t=n+1}^{T}\beta_t+8\sqrt{2nrT\log\left(\frac{T(T+1)}{\delta_T}\right)}+\sum_{t=n+1}^{T}2\left(2rC+\frac{n}{\beta_t}\right)n\delta_t\end{aligned}\tag{47}$$

For $t\in[1:n]$, define $\boldsymbol{p}(t)$ as $p_k(t)=1_{[k\in\mathcal{A}_1(t)]}$. This definition is consistent with the definition of $\boldsymbol{p}(t)$ for $t\in\{n+1,\dots\}$, since $\mathcal{A}_1(t)$ is deterministic for $t\in[1:n]$ (see lines 1-4 of Algorithm 1). Hence,

$$\mathbb{E}\left\{2nf^{\text{worst},*}-2\sum_{t=1}^{n}f^{\text{worst}}(\boldsymbol{p}(t))\right\}\le 2nf^{\text{worst},*}\le 2nrC,\tag{48}$$

where the last inequality follows from (43). Adding (47) and (48), and dividing by $2T$, we have that

$$\begin{aligned}&\mathbb{E}\left\{f^{\text{worst},*}-\frac{1}{T}\sum_{t=1}^{T}f^{\text{worst}}(\boldsymbol{p}(t))\right\}\le\frac{n}{2\beta_T T}+\frac{nrC}{T}+\frac{nD_T^2\sum_{t=n+1}^{T}\beta_t}{2T}\\&+4\sqrt{\frac{2nr\log\left(\frac{T(T+1)}{\delta_T}\right)}{T}}+\frac{1}{T}\sum_{t=n+1}^{T}\left(2rC+\frac{n}{\beta_t}\right)n\delta_t.\end{aligned}\tag{49}$$

Using the definition of $R^{\text{worst}}(T)$ defined in (7) in the above, we are done.

(b) To prove the (b), consider $\delta_t=\Theta(1/t)$ and $\beta_t=\Theta(1/\sqrt{t})$ for all $t\in\{n+1,n+2,\dots\}$. We analyze each term in the right hand side of the bound obtained in part-(a). Notice that $\frac{n}{2\beta_T T}$ is $\Theta(1/\sqrt{T})$, $\frac{nrC}{T}$ is $\Theta(1/T)$, and $\sqrt{\frac{2nr\log\left(\frac{T(T+1)}{\delta_T}\right)}{T}}$ is $\Theta(\sqrt{\log(T)/T})$. We will analyze the remaining two terms separately. For simplicity, we will use $\delta_t=1/t$ and $\beta_t=1/\sqrt{t}$. First,

$$\frac{nD_T^2\sum_{t=n+1}^{T}\beta_t}{2T}=_{(a)}\frac{n}{2T}\left(C+2\sqrt{2\log\left(T^2(T+1)\right)}\right)^2\sum_{t=n+1}^{T}\frac{1}{\sqrt{t}}=_{(b)}\Theta\left(\frac{\log(T)}{\sqrt{T}}\right),$$

where (a) follows from the definition of D_T in (28) and for (b) we have used $\sum_{k=1}^{l}1/\sqrt{k}\le 2\sqrt{l}$. Next,

$$\frac{1}{T}\sum_{t=n+1}^{T}\left(2rC+\frac{n}{\beta_t}\right)n\delta_t=\frac{1}{T}\sum_{t=n+1}^{T}\left(\frac{2rnC}{t}+\frac{n^2}{\sqrt{t}}\right)=_{(a)}\Theta\left(\frac{1}{\sqrt{T}}\right),$$

where for (a) we have used $\sum_{k=1}^{l}1/k\le\sum_{k=1}^{l}1/\sqrt{k}\le 2\sqrt{l}$. Combining the terms, we are done. □

3 Simulation Results

In this section we present our simulation results. In Fig. 2, we simulate the performance of our algorithm for $n = 6, m = 5, \boldsymbol{E} = [3, 1, 1, 1, 0.5, 0.1]$, and $r \in \{1, 2, 3\}$. For each value of *r*, we run Algorithm 1 for 2×10^6 iterations. We first plot $\frac{1}{t} \sum_{\tau=1}^{t} f^{\text{worst}}(\boldsymbol{p}(\tau))$ vs *t*, after which we plot the entries of $\boldsymbol{p}(t)$ vs *t*, where *t* is the iteration number. In the plots, we also plot the optimal objective value $f^{\text{worst},*}$, and the optimal $\boldsymbol{p}^*$ for reference. In Fig. 3, we repeat the above with parameters $n = 6, m = 5, \boldsymbol{E} = [6.1, 1, 1, 1, 0.5, 0.1]$ and $r \in \{1, 2, 3\}$.

Notice that in both cases, we use $\boldsymbol{E}$ sorted in nonincreasing order. Comparing the case $r = 1$ for the two values of $\boldsymbol{E}$ it can be seen that when $\boldsymbol{E} = [6.1, 1, 1, 1, 0.5, 0.1]$, player 1 chooses resource 1 with probability 1 while when $\boldsymbol{E} = [3, 1, 1, 1, 0.5, 0.1]$, player 1 chooses several resources with nonzero probability. This is because when $\boldsymbol{E} = [6.1, 1, 1, 1, 0.5, 0.1]$, the mean reward of the first resource is higher than five times the mean reward of the second resource. Hence, even if all the other players choose resource 1, player 1 will not benefit by choosing a different resource. From Fig. 2-Bottom-Left and Fig. 3-Bottom-Left, it can be seen that the online algorithm learns this behavior. However, when $r > 1$, player 1 chooses resource 1 with probability 1 for both values of $\boldsymbol{E}$. In all cases, it can be seen that the worst-case expected utility of the online algorithm converges to the optimal value.

Another interesting observation is the slower convergence of the algorithm for $r = 1$ with $\boldsymbol{E} = [6.1, 1, 1, 1, 0.5, 0.1]$. This may be due to the fact that this is the only case where the optimal solution $\boldsymbol{p}^*$ is an extreme point of $\Delta_{n,r}$ ($\boldsymbol{p}^*$ chooses all resources except resource 1 with zero probability). In particular, using $\boldsymbol{p}(t)$ close to $\boldsymbol{p}^*$ in the initial phases of the algorithm reduces exploration required to learn the E_k values.

4 Conclusions

In this paper, we considered the problem of worst-case time-average expected reward maximization for the first player in online multi-player resource-sharing games with bandit feedback. We considered a fair reward allocation model, where in each time slot, the reward of a resource is shared equally among the players selecting it. We provided an upper confidence bound algorithm that gets within $\mathcal{O}(\log(T)/\sqrt{T})$ of optimality within a finite time horizon of T time slots. Extending this work beyond the fair reward allocation model to general congestion games in the online setting is future work.

Appendix A: Madow's Sampling Technique

In this section, we present the Madow's sampling technique (Algorithm 2). The algorithm takes as an input a vector $\boldsymbol{p} \in \Delta_{n,r}$ and outputs a set $\mathcal{A} \subset [1 : n]$ such that $|\mathcal{A}| = r$, and $\mathbb{E}\{1_{k \in \mathcal{A}}\} = p_k$ for all $k \in [1 : n]$. See [1] for the proof of the correctness of the algorithm.

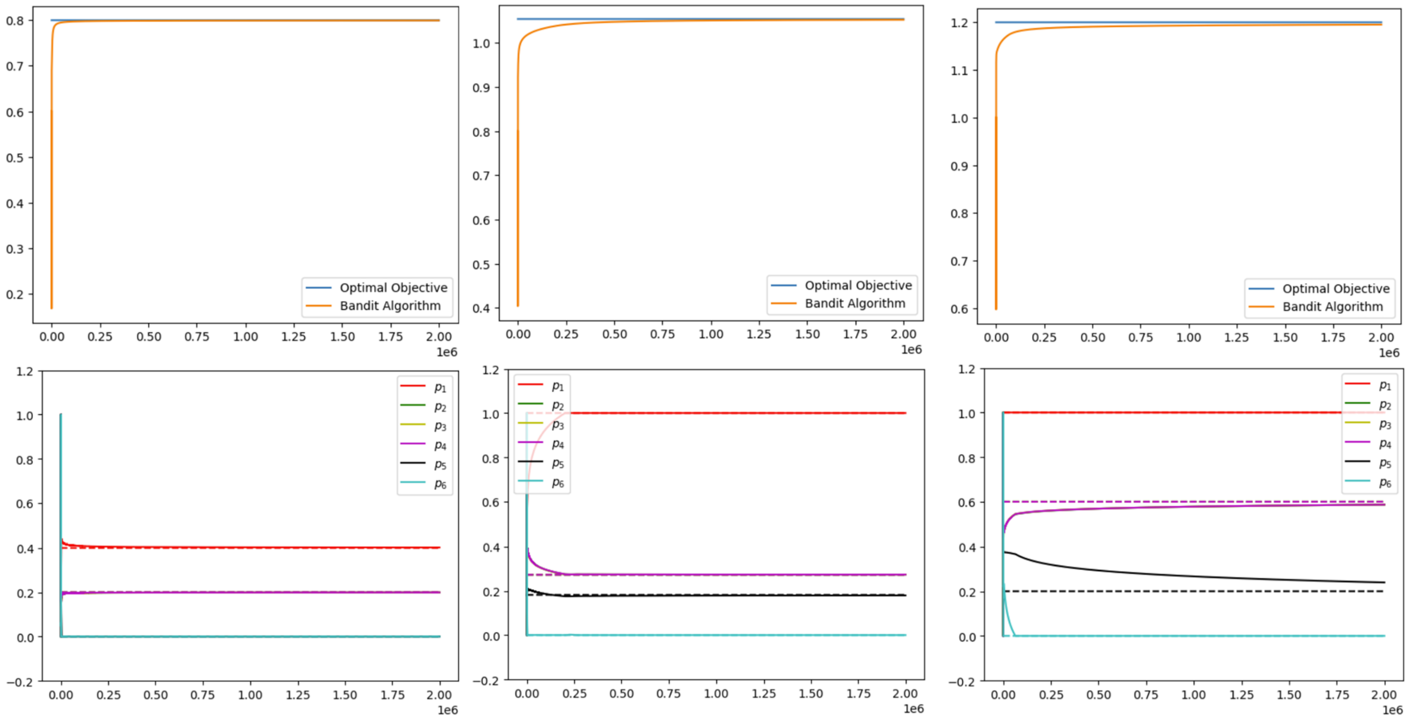

Fig. 2 Scenario $\boldsymbol{E} = [3, 1, 1, 1, 0.5, 0.1]$. Top: $\frac{1}{t}\sum_{\tau=1}^{t} f^{\text{worst}}(\boldsymbol{p}(\tau))$ and $f^{\text{worst},*}$ vs t, Bottom: Components of $\boldsymbol{p}(t)$ and components of $\boldsymbol{p}^*$ vs t for, Left: $r = 1$, Middle: $r = 2$, Right: $r = 3$

```
Define $\Pi_0 = 0$, and $\Pi_k = \Pi_{k-1} + p_k \; \forall k \in [1:n]$.
Sample $U \sim \text{Uniform}(0,1)$.
Define the set $\mathcal{S}_0 = \varnothing$, where $\varnothing$ denotes the empty set.
for each $k \in \{0, 1, \ldots, r-1\}$ do
    Find the unique $i \in [1:n]$ such that $\Pi_{i-1} \leq U + k < \Pi_i$.
    Define $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \{i\}$.
end
Output $\mathcal{A} = \mathcal{S}_r$.
```

Algorithm 2 Madow's sampling technique

Appendix B: Proof of Theorem 2

This section finds $\boldsymbol{p}^*$ for the case $m = 3, r = 1$ where n is a positive integer and $\boldsymbol{E} = [E_1, \ldots, E_n]$ is known. Recall that we use $\Delta_n = \Delta_{n,1}$. Define $\boldsymbol{p}^* \in \arg\min_{\boldsymbol{p} \in \Delta_n} f^{\text{worst}}(\boldsymbol{p})$, $f^{\text{worst}}(\boldsymbol{p}) = \min_{\boldsymbol{x} \in \mathcal{J}} f(\boldsymbol{p}, \boldsymbol{x})$, and $f(\boldsymbol{p}, \boldsymbol{x}) = \sum_{k=1}^{n} \frac{p_k E_k}{1 + x_k}$. Recall that we assumed without loss of generality that $\boldsymbol{E}$ is sorted as $E_k \geq E_{k+1}$ for all $k \in \{1, 2, \ldots, n-1\}$. Notice that from Lemma 1 we have,

$$f^{\text{worst}}(\boldsymbol{p}) = \begin{cases} \sum_{k=1}^{n} p_k E_k - \frac{2}{3}\Gamma_1(\boldsymbol{p}) & \text{if } \Gamma_1(\boldsymbol{p}) > 3\Gamma_2(\boldsymbol{p}) \\ \sum_{k=1}^{n} p_k E_k - \frac{1}{2}\Gamma_1(\boldsymbol{p}) - \frac{1}{2}\Gamma_2(\boldsymbol{p}) & \text{if } \Gamma_1(\boldsymbol{p}) \leq 3\Gamma_2(\boldsymbol{p}) \end{cases} \tag{A1}$$

where $\Gamma_1(\boldsymbol{p})$, $\Gamma_2(\boldsymbol{p})$ are the largest and the second largest elements of the set $\{p_k E_k; 1 \leq k \leq n\}$, respectively. Observe that if $\Gamma_1(\boldsymbol{p}) = 3\Gamma_2(\boldsymbol{p})$, then $\frac{2}{3}\Gamma_1(\boldsymbol{p}) = \frac{1}{2}\Gamma_1(\boldsymbol{p}) + \frac{1}{2}\Gamma_1(\boldsymbol{p})$. In particular, the function $f^{\text{worst}}(\boldsymbol{p})$ is continuous and so it has a maximizer $\boldsymbol{p}^*$ over the compact set Δ_n. By considering the case $\Gamma_1(\boldsymbol{p}^*) \geq 3\Gamma_2(\boldsymbol{p}^*)$ and a particular index $i \in \{1, \ldots, n\}$ achieves $p_i^* E_i = \Gamma_1(\boldsymbol{p}^*)$, and the case $\Gamma_1(\boldsymbol{p}^*) \leq 3\Gamma_2(\boldsymbol{p}^*)$ and particular indices $i \neq j$ achieve $p_i^* E_i = \Gamma_1(\boldsymbol{p}^*)$, $p_j^* E_j = \Gamma_2(\boldsymbol{p}^*)$, we notice that $\boldsymbol{p}^*$ is the solution of the problem with the maximal optimal objective out of the n^2 linear programs,

$$\begin{array}{lll} (\text{P1-}i): & \max & \sum\limits_{k=1}^{n} p_k E_k - \frac{2 p_i E_i}{3} \\ & \text{s.t.} & \boldsymbol{p} \in \Delta_n, \\ & & p_i E_i \geq 3 p_k E_k \;\; \forall 1 \leq k \leq n, \end{array} \tag{A2}$$

and

$$\begin{array}{lll} (\text{P1-}(i,j)): & \max & \sum\limits_{k=1}^{n} p_k E_k - \frac{p_i E_i}{2} - \frac{p_j E_j}{2} \\ & \text{s.t.} & \boldsymbol{p} \in \Delta_n, \;\; p_i E_i \leq 3 p_j E_j, \;\; p_i E_i \geq p_j E_j, \\ & & p_j E_j \geq p_k E_k \;\; \forall 1 \leq k \leq n, k \neq i, \end{array} \tag{A3}$$

where $i, j \in [1:n]$ and $i \neq j$. To solve (P1-i), and (P1-(i, j)), it shall be useful to re-index to associate i with 1, and (i, j) with 1 and 2. Hence, we define the two problems.

$$\begin{array}{lll}\text{(P1-1)}: & \max & f_1(\boldsymbol{p}) = \sum\limits_{k=1}^{n} p_k F_k - \frac{2p_1F_1}{3} \\ & \text{s.t.} & \boldsymbol{p} \in \Delta_n, \\ & & p_1F_1 \geq 3p_{k+1}F_{k+1} \ \forall k \in \{1, \ldots, n-1\},\end{array} \tag{A4}$$

and

$$\begin{array}{lll}\text{(P1-2)}: & \max & f_2(\boldsymbol{p}) = \sum\limits_{k=1}^{n} p_k F_k - \frac{p_1F_1}{2} - \frac{p_2F_2}{2} \\ & \text{s.t.} & \boldsymbol{p} \in \Delta_n, \, p_1F_1 \leq 3p_2F_2, \ p_1F_1 \geq p_2F_2, \\ & & p_2F_2 \geq p_kF_k \ \forall 3 \leq k \leq n,\end{array} \tag{A5}$$

where for (P1-1), without loss of generality $\boldsymbol{F} \in \mathbb{R}^n$ is assumed to a positive vector such that $F_k \geq F_{k+1}$ for $k \in [2 : n-1]$, and for (P1-2), $\boldsymbol{F} \in \mathbb{R}^n$ is assumed to a positive vector such that $F_k \geq F_{k+1}$ for $k \in [3 : n-1]$. It should be noted that the F_k values are just the E_k values under more convenient indexing. Solving the above two problems immediately solves each of the previously defined n^2 problems. Define the two sequences $(U_i; 1 \leq i \leq n)$, and $(V_i; 2 \leq i \leq n)$ by,

$$U_i = \frac{i}{\frac{3}{F_1} + \sum_{k=2}^{i} \frac{1}{F_k}}, \tag{A6}$$

and,

$$V_i = \frac{i-1}{\sum_{k=1}^{i} \frac{1}{F_k}}. \tag{A7}$$

These two sequences are useful when constructing the solutions to (P1-1) and (P1-2).

We first state a lemma that is useful for the proof.

Lemma 4 Consider constrained optimization problem

$$\begin{array}{lll}\underset{\boldsymbol{x} \in \mathcal{Y}}{\max} & z_0(\boldsymbol{x}) & \\ \text{s.t.} & z_i(\boldsymbol{x}) & \geq 0 \ \text{ for } i \in \{1, 2, \ldots, k\},\end{array} \tag{A8}$$

where $z_i : \mathbb{R}^n \to \mathbb{R}$ for $i \in \{0, 1, 2, \ldots, k\}$, and $\mathcal{Y} \subset \mathbb{R}^n$. Consider the unconstrained problem $\max_{\boldsymbol{x} \in \mathcal{Y}} z_0(\boldsymbol{x}) + \sum_{i=1}^{k} \mu_i z_i(\boldsymbol{x})$ for some $\boldsymbol{\mu} \geq 0$. Let $\boldsymbol{x}^*$ be a solution to the unconstrained problem. Assume $\boldsymbol{x}^*$ satisfies for all $i \in \{1, 2, \ldots, k\}$,

(a) $z_i(\boldsymbol{x}^*) \geq 0$ (That is $\boldsymbol{x}^*$ is feasible for the constrained problem)
(b) $\mu_i > 0$ implies $z_i(\boldsymbol{x}^*) = 0$.

Then $\boldsymbol{x}^*$ is optimal for the constrained problem.

Proof The proof of the lemma is immediate and omitted for brevity.

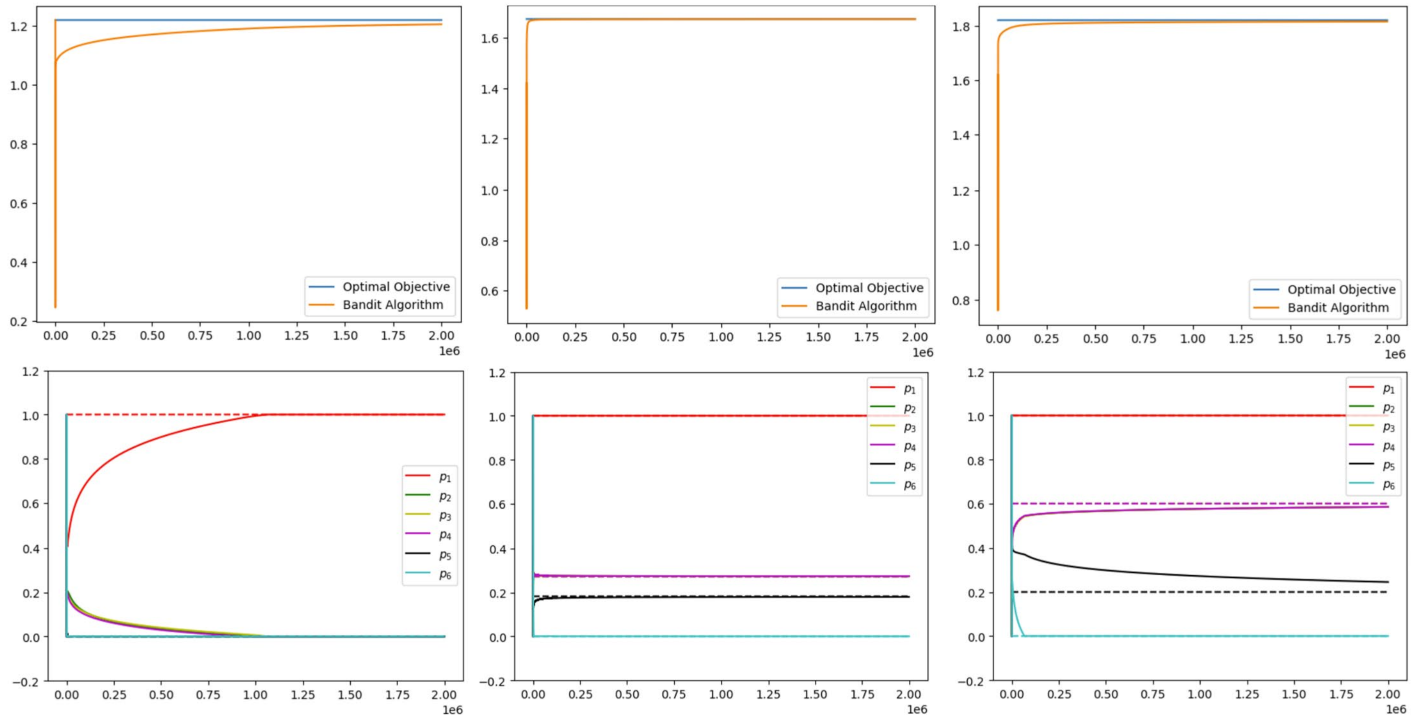

Fig. 3 Scenario $\boldsymbol{E} = [6.1, 1, 1, 1, 0.5, 0.1]$. Top: $\frac{1}{t}\sum_{\tau=1}^{t} f^{\text{worst}}(\boldsymbol{p}(\tau))$ and $f^{\text{worst},*}$ vs t, Bottom: Components of $\boldsymbol{p}(t)$ and components of $\boldsymbol{p}^*$ vs t, Left: $r = 1$, Middle: $r = 2$, Right: $r = 3$

B.0.1 Solving (P1-1)

Consider the problem (P1-1):

$$\begin{array}{lll}\text{(P1-1)}: & \max & f_1(\boldsymbol{p}) \\ & \text{s.t.} & \boldsymbol{p}\in \Delta_n, \\ & & p_1F_1\geq 3p_{k+1}F_{k+1}\ \forall k\in\{1,2,\ldots,n-1\},\end{array} \tag{A9}$$

where the function f_1 is defined by

$$f_1(\boldsymbol{p})=\sum_{k=1}^{n}p_kF_k-\frac{2p_1F_1}{3}. \tag{A10}$$

Let us define

$$u=\arg\max_{1\leq i\leq n}U_i, \tag{A11}$$

where the sequence $(U_i;1\leq i\leq n)$ is defined in (A6) and $\arg\max$ returns the least index in the case of ties. We establish that the solution to (P1-1) is $\tilde{\boldsymbol{p}}^*$, where

$$\tilde{p}_k^*=\begin{cases}\frac{\frac{3}{F_1}}{\frac{3}{F_1}+\sum_{j=2}^{u}\frac{1}{F_j}} & \text{if } k=1\\ \frac{\frac{1}{F_k}}{\frac{3}{F_1}+\sum_{j=2}^{u}\frac{1}{F_j}} & \text{if } 2\leq k\leq u\\ 0 & \text{otherwise,}\end{cases} \tag{A12}$$

with optimal objective value U_u.
Consider the vector $\tilde{\boldsymbol{\mu}}^*\in\mathbb{R}^{n-1}$ defined by

$$\tilde{\mu}_k^*=\begin{cases}\frac{1}{3}\left(1-\frac{1}{F_{k+1}}\frac{u}{\frac{3}{F_1}+\sum_{j=2}^{u}\frac{1}{F_j}}\right) & \text{if } 1\leq k\leq u-1\\ 0 & \text{otherwise,}\end{cases} \tag{A13}$$

where u is defined in (A11). In the subsequent analysis, we establish that $\tilde{\boldsymbol{\mu}}^*$ defined above is a valid Lagrange multiplier ($\tilde{\mu}_k^*\geq 0$ for all $k\in[1:n-1]$) and $(\tilde{\boldsymbol{p}}^*,\tilde{\boldsymbol{\mu}}^*)$ satisfy the conditions of Lemma 4, where for $k\in[1:n-1]$, $\tilde{\mu}_k^*$ corresponds to the constraint $p_1F_1\geq 3p_{k+1}F_{k+1}$ of (P1-1). This establishes that $\tilde{\boldsymbol{p}}^*$ solves (P1-1). It can be easily checked by substitution that the objective value of (P1-1) for $\tilde{\boldsymbol{p}}^*$ is U_u. Hence, the steps of the proof can be summarized as:

1. $\tilde{\mu}_k^*\geq 0$ for all $k\in[1:n-1]$.
2. $\tilde{\boldsymbol{p}}^*$ is feasible for (P1-1). In particular, we have that $\tilde{\boldsymbol{p}}^*\in\Delta_n$ and $\tilde{p}_1^*F_1\geq 3\tilde{p}_{k+1}^*F_{k+1}$ for $k\in\{1,\ldots,n-1\}$.
3. $\tilde{\boldsymbol{p}}^*$ solves the unconstrained problem with Lagrange multiplier vector $\tilde{\boldsymbol{\mu}}^*$ (See Lemma 4 for the construction of the unconstrained problem).

4. For $k \in \{1, \ldots, n-1\}$, $\tilde{\mu}_k^* > 0$ implies the corresponding constraint of (P1-1) is met with equality.

Notice that step 2 above can be checked by direct substitution from (A12). Also, for step 4, notice that from the definition of $\tilde{\boldsymbol{\mu}}^*$ in (A13), $\tilde{\mu}_k^* > 0$ implies that $k \in \{1, \ldots, u-1\}$. By substitution from the definition of $\tilde{\boldsymbol{p}}^*$ in (A12), it follows that $\tilde{p}_1^* F_1 = 3\tilde{p}_{k+1}^* F_{k+1}$ for $k \in \{1, \ldots, u-1\}$. Hence, we are only required to establish steps 1 and 3. We establish step 1 along with two other results that will be useful for step 3 in Lemma 5 below, after which we establish step 3 in Lemma 6.

Lemma 5 Consider the $\tilde{\boldsymbol{\mu}}^*$ defined in (A13). We have that

(a) $\tilde{\mu}_k^* \geq 0$ for all k such that $1 \leq k \leq n-1$.
(b) $F_k(1 - 3\tilde{\mu}_{k-1}^*) = \frac{u}{\frac{3}{F_1} + \sum_{i=2}^{u} \frac{1}{F_i}}$ for $2 \leq k \leq u$ and
$F_1 \left(\frac{1}{3} + \sum_{i=1}^{u-1} \tilde{\mu}_i^* \right) = \frac{u}{\frac{3}{F_1} + \sum_{i=2}^{u} \frac{1}{F_i}}$.
(c) $F_k \leq \frac{u}{\frac{3}{F_1} + \sum_{j=2}^{u} \frac{1}{F_j}}$ for $u + 1 \leq k \leq n$.

Proof Notice that since $u = \arg\max_{1 \leq i \leq n} U_i$, we have that

$$U_u \geq U_j \text{ for all } j \in [1:n]. \tag{A14}$$

(a) Notice that when $k > u - 1$, by definition of $\tilde{\boldsymbol{\mu}}^*$ in (A13), we have that $\tilde{\mu}_k^* = 0$. Now suppose $k \leq u - 1$. Hence, we can assume $u \geq 2$. From the definition of $\tilde{\boldsymbol{\mu}}^*$ in (A13), we are required to prove $F_{k+1} \geq \frac{u}{\frac{3}{F_1} + \sum_{j=2}^{u} \frac{1}{F_j}}$ for all $k \in \{1, 2, \ldots, u-1\}$. It is enough to prove the above for $k = u - 1$, since $F_k \geq F_{k+1}$ for $k \geq 2$. Notice that from (A14) we have that $U_u \geq U_{u-1}$ (recall that $u \geq 2$). Substituting from (A6), $U_u \geq U_{u-1}$ translates to $\frac{u}{\frac{3}{F_1} + \sum_{j=2}^{u} \frac{1}{F_j}} \geq \frac{u-1}{\frac{3}{F_1} + \sum_{j=2}^{u-1} \frac{1}{F_j}}$. Simplifying the above gives $F_u \geq \frac{u}{\frac{3}{F_1} + \sum_{j=2}^{u} \frac{1}{F_j}}$ as desired.
(b) Substituting from the definition of $\tilde{\mu}_k^*$ in (A13) and simplifying yields the result.
(c) If $u = n$, there is nothing to prove. Hence, we can assume $u < n$. Notice that it is enough to prove the result for $k = u + 1$, since $F_k \geq F_{k+1}$ for $k \geq 2$. From (A14), we have that $U_u \geq U_{u+1}$ (recall that $u < n$). Substituting from (A6), $U_u \geq U_{u+1}$ translates to $\frac{u}{\frac{3}{F_1} + \sum_{j=2}^{u} \frac{1}{F_j}} \geq \frac{u+1}{\frac{3}{F_1} + \sum_{j=2}^{u+1} \frac{1}{F_j}}$. Simplifying the above we have $F_{u+1} \leq \frac{u}{\frac{3}{F_1} + \sum_{j=2}^{u} \frac{1}{F_j}}$ as desired.□

Lemma 6 The vector $\tilde{\boldsymbol{p}}^*$ defined in (A12) solves unconstrained problem with Lagrange multiplier vector $\tilde{\boldsymbol{\mu}}^*$ defined in (A13) (See Lemma 4 for the construction of the unconstrained problem). In particular, $\tilde{\boldsymbol{p}}^*$ solves

$$\begin{array}{ll} \max & f_1(\boldsymbol{p}) + \sum\limits_{k=1}^{n-1} \tilde{\mu}_k^*(p_1F_1 - 3p_{k+1}F_{k+1}) \\ \text{s.t.} & \boldsymbol{p} \in \Delta_n, \end{array} \tag{A15}$$

where the function f_1 is defined in (A10).

Proof Noticing from the definition of $\tilde{\boldsymbol{\mu}}^*$ in (A13) that $\tilde{\mu}_k^* = 0$ for $k > u$, and using the definition of function f_1 in (A10), the objective of the above unconstrained problem simplifies as

$$f_1(\boldsymbol{p}) + \sum_{k=1}^{n-1} \tilde{\mu}_k^*(p_1F_1 - 3p_{k+1}F_{k+1}) = p_1F_1\left(\frac{1}{3} + \sum_{i=1}^{u-1}\tilde{\mu}_i^*\right) + \sum_{k=2}^{u} p_kF_k(1 - 3\tilde{\mu}_{k-1}^*)$$
$$+ \sum_{k=u+1}^{n} p_kF_k = \sum_{i=1}^{u} p_iC + \sum_{k=u+1}^{n} p_kF_k,$$

where $C = \frac{u}{\frac{3}{F_1} + \sum_{i=2}^{u}\frac{1}{F_i}}$ and the last equality follows from Lemma 5-(b). Also, notice that from Lemma 5-(c), we have that $C \geq F_k$ for all $k \in \{u+1, \ldots, n\}$. Hence, the optimal solution to the above defined unconstrained problem is any $\boldsymbol{p} \in \Delta_n$ such that $p_k = 0$ for all $k \in \{u+1, \ldots, n\}$. In particular, $\tilde{\boldsymbol{p}}^*$ given in (A12) is a solution to the unconstrained problem. □

B.0.2 Solving (P1-2)

Consider the problem (P1-2).

$$\begin{array}{lll} \text{(P1-2)}: & \max & f_2(\boldsymbol{p}) \\ & \text{s.t.} & \boldsymbol{p} \in \Delta_n,\ p_1F_1 \leq 3p_2F_2,\ p_1F_1 \geq p_2F_2 \\ & & p_2F_2 \geq p_kF_k\ \forall 3 \leq k \leq n, \end{array} \tag{A16}$$

where the function f_2 is defined as

$$f_2(\boldsymbol{p}) = \sum_{k=1}^{n} p_kF_k - \frac{p_1F_1}{2} - \frac{p_2F_2}{2} \tag{A17}$$

Let us define $u = \arg\max_{2 \leq i \leq n} U_i$ and $v = \arg\max_{2 \leq i \leq n} V_i$ where the sequences $(U_i; 1 \leq i \leq n)$, and $(V_i; 2 \leq i \leq n)$ are defined in (A6), and (A7), respectively, and $\arg\max$ returns the least index in the case of ties. In this case, to define *u*, we only consider the indices of the $(U_i; 1 \leq i \leq n)$ sequence starting from 2 in contrast to the definition of *u* in the solution to (P1-1). The solution of (P1-2) can be described under two cases.
Case 1: $V_v > U_u$: The solution to (P1-2) in this case is $\hat{\boldsymbol{p}}^*$ where

$$\hat{p}_k^* = \begin{cases} \frac{\frac{1}{F_k}}{\sum_{j=1}^{v} \frac{1}{F_j}} & \text{if } 1 \leq k \leq v \\ 0 & \text{otherwise,} \end{cases} \tag{A18}$$

with optimal objective value V_v.

Case 2: $V_v \leq U_u$: The solution to (P1-2) in this case is $\bar{\boldsymbol{p}}^*$ where

$$\bar{p}_k^* = \begin{cases} \frac{\frac{3}{F_1}}{\frac{3}{F_1} + \sum_{j=2}^{u} \frac{1}{F_j}} & \text{if } k = 1 \\ \frac{\frac{1}{F_k}}{\frac{3}{F_1} + \sum_{j=2}^{u} \frac{1}{F_j}} & \text{if } 2 \leq k \leq u \\ 0 & \text{otherwise.} \end{cases} \tag{A19}$$

with optimal objective value U_u.

The proof is similar to the Solution of (P1-1). We omit the proof for brevity. For the complete proof, refer the technical report [52].

B.0.3 Finding $\boldsymbol{p}^*$

Finally, we are ready to combine the solutions of (P1-1) and (P1-2) to find $\boldsymbol{p}^* \in \arg\max_{\boldsymbol{p} \in \Delta_n} f^{\text{worst}}(\boldsymbol{p})$. Notice that since we solved (P1-1) and (P1-2), we have solved all of the n^2 problems (P1-i), and (P1-(i,j)) for $i,j \in [1:n]$ such that $i \neq j$ defined in (A2) and (A3), respectively. Hence, we can find $\boldsymbol{p}^*$ by solving all the above problems and finding the one that gives the highest optimal objective. But, it turns out that it is, in fact, enough to solve (P1-1), and (P1-$(1,2)$). To prove this, consider arbitrary (i, j) such that $1 \leq i,j \leq n$ such that $i \neq j$. Define, $\boldsymbol{D} \in \mathbb{R}^n$ to be the vector obtained by permuting the entries of $\boldsymbol{E}$ such that $D_1 = E_i, D_2 = E_j$, and $D_k \geq D_{k+1}$ for $k \in [3 : n-1]$. Notice that due to the solution of (P1-2), the optimal value of (P1-(i, j)) is given by $\gamma^* = \max \left\{ \frac{a-1}{\sum_{k=1}^{a} \frac{1}{D_k}}, \frac{b}{\frac{3}{D_1} + \sum_{k=2}^{b} \frac{1}{D_k}} \middle| 2 \leq a,b \leq n \right\}$. Notice that,

$$\max \left\{ \frac{a-1}{\sum_{k=1}^{a} \frac{1}{E_k}}, \frac{b}{\frac{3}{E_1} + \sum_{k=2}^{b} \frac{1}{E_k}} \middle| a,b \in [2:n] \right\} \geq \gamma^*, \tag{A20}$$

where the inequality follows since $\sum_{k=1}^{a} \frac{1}{E_k} \leq \sum_{k=1}^{a} \frac{1}{D_k}$, and $\frac{3}{E_1} + \sum_{k=2}^{b} \frac{1}{E_k} \leq \frac{3}{D_1} + \sum_{k=2}^{b} \frac{1}{D_k}$ for all $a, b \in [2:n]$. This follows since $E_k \geq E_{k+1}$ for all $k \in [1 : n-1]$. But notice that the left-hand side of (A20) is the optimal value of (P1-(1, 2)). Hence, the optimal value of (P1-(1, 2)) is at least as that of (P1-(i, j)). Hence, it is enough to solve (P1-(1, 2)). With similar reasoning, we can establish that solving (P1-1) suffices. Considering the solutions (P1-(1, 2)) and (P1-1), we have the result.

Appendix C

Given $\boldsymbol{F} \in \mathbb{R}_{+}^{n}$, and $\boldsymbol{p} \in \Delta_{n,r}$, we focus on finding $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathcal{J}} \sum_{k=1}^{n} \frac{p_k F_k}{1+x_k}$. This is an optimization over a nonconvex discrete set $\boldsymbol{x} \in \mathcal{J}$. However, it has a classical separable structure that is well studied in the literature and can be solved exactly using either a greedy $\mathcal{O}(n + mr\log(n))$ incremental algorithm or an improved $\mathcal{O}(n\log(mr))$ algorithm. For completeness, we summarize an $\mathcal{O}(nmr)$ algorithm in Algorithm 3. For improved algorithms, refer to the work of [53].

```
Initialize $\boldsymbol{x} = [0, 0, \ldots, 0] \in \mathbb{N}^n$.
for each iteration $k \in [1 : (m-1)r]$ do
    Increase $x_i$ by 1 where $i \in \arg\min_{\substack{k\in[1:n] \\ x_k < m-1}} \left\{ \frac{p_k F_k}{1+x_k} - \frac{p_k F_k}{2+x_k} \right\}$.
end
Output $\boldsymbol{x}$.
```

Algorithm 3 Algorithm for Appendix C

Appendix D: Algorithm to Project onto $\Delta_{n,r}$

```
Define for all $1 \le a \le b \le n$,
    $\mu_{a,b} = \frac{\sum_{j=a}^{b} y_j - (r-a+1)}{b-a+1}, \mathcal{A}_{a,b} = \mathbb{1}\{y_b \ge \mu_{a,b} \ge y_a - 1\}$
    $\mathcal{B}_{a,b} = \mathbb{1}\{(b = n) \text{ or } [(b < n) \text{ and } (y_{b+1} < \mu_{a,b})]\}$
    $\mathcal{C}_{a,b} = \mathbb{1}\{(a = 1) \text{ or } [(a > 1) \text{ and } (y_{a-1} - 1 > \mu_{a,b})]\}$
    $g(a,b) = \min\{c : c \ge b, \mathcal{B}_{a,c} = 1\}, h(a,b) = \max\{c : c \le a, \mathcal{C}_{c,b} = 1\}$.
Initialize $(a_1, b_1) = (r, r)$.
for each $t \in \{1, 2, \ldots\}$ do
    Set $(a_{t+1}, b_{t+1}) = (h(a_t, g(a_t, b_t)), g(a_t, b_t))$.
    if $(a_{t+1}, b_{t+1}) = (a_t, b_t)$ then
        Output $\boldsymbol{x} \in \mathbb{R}^n$, where $x_i = \Pi_{[0,1]}(y_i - \mu_{a_t,b_t})$.
    end
end
```

Algorithm 4 Projecting $\boldsymbol{y}$ sorted in the nonincreasing order onto $\Delta_{n,r}$

Analysis of Algorithm 4: Fix $\boldsymbol{y} \in \mathbb{R}$. Notice that the problem of projection of $\boldsymbol{y} \in \mathbb{R}^n$ onto $\Delta_{n,r}$ is,

$$\begin{array}{ll} \min\limits_{\boldsymbol{z}} & \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|^2 \\ \text{s.t} & \boldsymbol{y} \in \Delta_{n,r} \end{array} \tag{A21}$$

We assume, without loss of generality, that $\boldsymbol{y}$ is sorted in non-increasing order (Notice that if $\boldsymbol{y}$ is not sorted, we could sort $\boldsymbol{y}$, perform the projection, and rearrange the elements according to the original order. This works since the set $\Delta_{n,r}$ is closed under the permutation of entries of its element vectors).

Now consider $L(\boldsymbol{z}, \mu)$ for $\mu \in \mathbb{R}$ given by $L(\boldsymbol{z}, \mu) = \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|^2 + \mu \left(\sum_{j=1}^n z_j - r\right)$, and the problem,

$$\begin{array}{lll} (\text{P6-}\mu) & \min\limits_{\boldsymbol{z}} & L(\boldsymbol{z}, \mu) \\ \text{s.t} & \boldsymbol{z} \in [0,1]^n & \end{array} \tag{A22}$$

for a fixed $\mu \in \mathbb{R}$. Let us assume the existence of a $\mu^* \in \mathbb{R}$ such that the solution $\boldsymbol{z}^*$ of (P6-μ^*) defined in (A22) satisfies, $\sum_{j=1}^n z_j^* = r$. Notice that $\boldsymbol{z}^*$ is optimal for the original problem since for any $\boldsymbol{z} \in \Delta_{n,r}$,

$$\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|^2 = L(\boldsymbol{z}, \mu^*) \geq L(\boldsymbol{z}^*, \mu^*) = \frac{1}{2}\|\boldsymbol{z}^* - \boldsymbol{y}\|^2.$$

Hence, we focus on finding such a μ^* and the corresponding $\boldsymbol{z}^*$. First, we focus on solving (P6-μ) defined in (A22) for a fixed $\boldsymbol{\mu} \in \mathbb{R}$. Notice that (P6-$\mu$) is a separable quadratic program in the entries of $\boldsymbol{z}$. Hence, the optimal z_j can be obtained by projecting the unconstrained optimal value for each entry of $\boldsymbol{z}$ onto [0, 1]. Hence, the solution is $z_j = \Pi_{[0,1]}(y_j - \mu)$ for all $j \in [1:n]$, where $\Pi_{[0,1]}$ denotes the projection operator onto [0, 1].

Now we need to find μ^* such that the optimal solution $\boldsymbol{z}^*$ of (P6-μ^*) defined in (A22) satisfies $\boldsymbol{z}^* \in \Delta_{n,r}$. Hence, we require $\mu^* \in \mathbb{R}$ such that

$$\sum_{j=1}^n \Pi_{[0,1]}(y_j - \mu^*) = r. \tag{A23}$$

For $\mu \in \mathbb{R}$, define the set $\mathcal{K}_\mu = \{i; 1 \leq i \leq n, \mu + 1 \geq y_i \geq \mu\}$. Notice that for each $\mu \in \mathbb{R}$, $\mathcal{K}_\mu$ is either the empty set or a set of the form $[a : b]$ where $1 \leq a \leq b \leq n$.

We have two possibilities if $\mathcal{K}_{\mu^*}$ is the empty set. The first is $\mu^* > y_j$ for all $j \in [1:n]$ in which case we have $\sum_{j=1}^n \Pi_{[0,1]}(y_j - \mu^*) = 0$ which does not agree with (A23). The second is $\mu^* < y_j - 1$ for all $j \in [1:n]$ in which case we have $\sum_{j=1}^n \Pi_{[0,1]}(y_j - \mu^*) = n$. This is only possible when $n = r$, in which case the only solution to the problem is the trivial solution of player 1 choosing all the resources.

Hence, we will focus on the case of non-empty K_{μ^*}. Let $\mathcal{K}_{\mu^*} = [a^* : b^*]$ where $1 \leq a^* \leq b^* \leq n$. This is equivalent to μ^* satisfying the conditions,

$$\begin{aligned} & y_{b^*} \geq \mu^* \geq y_{a^*} - 1 \\ & (b^* = n) \text{ or } [(b^* < n) \text{ and } (y_{b^*+1} < \mu^*)] \\ & (a^* = 1) \text{ or } [(a^* > 1) \text{ and } (y_{a^*-1} - 1 > \mu^*)] \end{aligned} \tag{A24}$$

Define for each $a, b \in [1:n]$ the real number $\mu_{a,b}$ as

$$\mu_{a,b} = \frac{\sum_{j=a}^b y_j - (r - a + 1)}{b - a + 1}. \tag{A25}$$

Now, notice that (A23) translates to,

$$\mu^* = \mu_{a^*,b^*}, \tag{A26}$$

where $\mathcal{K}_{\mu^*} = [a^* : b^*]$. Combining (A26) and (A24), we have that if we can find a^*, b^* $(1 \leq a^* \leq b^* \leq n)$ such that

$$\begin{aligned}
&y_{b^*} \geq \mu_{a^*,b^*} \geq y_{a^*} - 1 \\
&(b^* = n) \text{ or } [(b^* < n) \text{ and } (y_{b^*+1} < \mu_{a^*,b^*})] \\
&(a^* = 1) \text{ or } [(a^* > 1) \text{ and } (y_{a^*-1} - 1 > \mu_{a^*,b^*})]
\end{aligned}$$

are all satisfied, then we are guaranteed that the solution $\boldsymbol{z}^*$ of (P6-μ_{a^*,b^*}) defined in (A22) satisfies $\boldsymbol{z}^* \in \Delta_{n,r}$. For each $a, b \in [1 : n]$, we will denote the three conditions,

$$\begin{aligned}
&\mathcal{A}_{a,b} = 1\{y_b \geq \mu_{a,b} \geq y_a - 1\} \\
&\mathcal{B}_{a,b} = 1\{(b = n) \text{ or } [(b < n) \text{ and } (y_{b+1} < \mu_{a,b})]\} \\
&\mathcal{C}_{a,b} = 1\{(a = 1) \text{ or } [(a > 1) \text{ and } (y_{a-1} - 1 > \mu_{a,b})]\}
\end{aligned}$$

Hence, our goal is to find (a^*, b^*) such that $\mathcal{A}_{a^*,b^*} = 1$, $\mathcal{B}_{a^*,b^*} = 1$, and $\mathcal{C}_{a^*,b^*} = 1$.

An easy way to find a^*, b^* is to go through all $a, b \in [1 : n]$ and check whether the above three conditions are satisfied. This approach has to go through n^2 pairs (*a*, *b*). We will provide an alternative approach that is efficient and goes through at most *n* pairs (*a*, *b*). With this approach, we can also establish the existence of $a^*, b^* \in [1 : n]$ satisfying $\mathcal{A}_{a^*,b^*} = 1$, $\mathcal{B}_{a^*,b^*} = 1$, and $\mathcal{C}_{a^*,b^*} = 1$.

Given $a, b \in [1 : n]$, define *g*(*a*, *b*) as the minimum integer in [*b* : *n*] such that $\mathcal{B}_{a,g(a,b)} = 1$ (Notice that $\mathcal{B}_{a,n} = 1$, so such an integer always exists). Similarly, define *h*(*a*, *b*) as the maximum integer in [1 : *a*] such that $\mathcal{C}_{h(a,b),b} = 1$ (Notice that $\mathcal{C}_{1,b} = 1$, so such an integer always exists).

We have the following claim.

Claim 1: *If* $\mathcal{A}_{a,b} = 1$ *then we have that* $\mathcal{A}_{a,g(a,b)} = 1$ *and* $\mathcal{A}_{h(a,b),b} = 1$

Proof We only prove that $\mathcal{A}_{a,g(a,b)} = 1$. The other part follows from a similar argument. First, notice that if $g(a, b) = b$, we are done. Hence, we will assume $g(a, b) > b$. We prove a stronger statement. We prove that $\mathcal{A}_{a,c} = 1$ for all $c \in [b : g(a, b)]$. We use induction for the proof. Notice that the base case $c = b$ is true. Now assume that $\mathcal{A}_{a,c} = 1$ for some $c \in [b : g(a, b) - 1]$. We prove that $\mathcal{A}_{a,c+1} = 1$. Since $c \in [b : g(a, b) - 1]$, from the definition of function *g*, we have that $\mathcal{B}_{a,c} = 0$. Also since $c \leq g(a, b) - 1$, we have that $c < n$. Hence, using the definition of $\mathcal{B}_{a,c}$, we have that $y_{c+1} \geq \mu_{a,c}$. Hence,

$$\mu_{a,c+1} = \frac{\mu_{a,c}(c - a + 1) + y_{c+1}}{c - a + 2} \leq \frac{y_{c+1}(c - a + 1) + y_{c+1}}{c - a + 2} = y_{c+1},$$

where for the first equation we have used the definition of $\mu_{a,c+1}$ from (A25). Also,

$$\mu_{a,c+1} = \frac{\mu_{a,c}(c-a+1)+y_{c+1}}{c-a+2} = \mu_{a,c} + \frac{y_{c+1}-\mu_{a,c}}{c-a+2} \geq_{(a)} \mu_{a,c} \geq_{(b)} y_a - 1,$$

where (a) follows since $y_{c+1} \geq \mu_{a,c}$ and (b) follows since $\mathcal{A}_{a,c}$ is true by assumption. From the above two inequalities, we have that $\mathcal{A}_{a,c+1} = 1$ as desired. □

Now consider the following sequence $\mathcal{S}$ of tuples $\mathcal{S} = \{(a_1, b_1), (a_2, b_2), \dots\}$, where $(a_1, b_1) = (r, r)$, and $(a_i, b_i) = (h(a_{i-1}, g(a_{i-1}, b_{i-1})), g(a_{i-1}, b_{i-1}))$ for each $i > 1$. We have the following claim regarding $\mathcal{S}$.

Claim 2: *We have that $\mathcal{A}_{a_i,b_i} = 1$ and $\mathcal{C}_{a_i,b_i} = 1$ for all $i \in \{2, 3, \dots\}$.*

Proof The fact that $\mathcal{C}_{a_i,b_i} = 1$ for all $i \in \{2, 3, \dots\}$ follows from the definition of a_i, b_i and the function *h*, since $(a_i, b_i) = (h(a_{i-1}, g(a_{i-1}, b_{i-1})), g(a_{i-1}, b_{i-1}))$ for all $i > 1$. For the other part we use induction. It can be easily checked that $\mathcal{A}_{a_1,b_1} = \mathcal{A}_{r,r} = 1$. Assume $\mathcal{A}_{a_i,b_i} = 1$ for some $i \geq 1$. Hence, we have from claim 1 that $\mathcal{A}_{a_i,g(a_i,b_i)} = 1$. Applying claim 1 again we have that $\mathcal{A}_{h(a_i,g(a_i,b_i)),g(a_i,b_i)} = 1$ which completes the induction. □

Now notice that the sequence $\mathcal{S}$ satisfies,

$$a_{i+1} \leq a_i, b_{i+1} \geq b_i \tag{A27}$$

for all $i \in \{1, 2, \dots\}$. This is because $b_{i+1} = g(a_i, b_i) \geq b_i$ by definition of function *g* and $a_{i+1} = h(a_i, g(a_i, b_i)) \leq a_i$ by definition of function *h*. Additionally, from the definition of sequence $\mathcal{S}$, it can be easily seen that if $(a_{i+1}, b_{i+1}) = (a_i, b_i)$ for some $i \geq 1$, then we have $(a_j, b_j) = (a_i, b_i)$ for all $j \geq i$. Combining the above property with (A27), we have that the sequence $\mathcal{S}$ is eventually constant. In particular, there exists $i \geq 1$ such that $(a_j, b_j) = (\bar{a}, \bar{b})$ for all $j \geq i$. It is also not difficult to see that the minimum such *i* satisfies $i \leq n$. To see this, notice that,

$$n - 1 \geq b_i - a_i = \sum_{j=1}^{i-1} [b_{j+1} - b_j + a_j - a_{j+1}] \geq (i-1), \tag{A28}$$

where the last inequality follows since for each $j < i$, we should have $a_{j+1} \leq a_j$ and $b_{j+1} \geq b_j$, and at least one of the two inequalities is strict (if not we will have $(a_{j+1}, b_{j+1}) = (a_j, b_j)$ which will contradict the minimality of *i*).

From claim 2 we have that $\mathcal{A}_{\bar{a},\bar{b}} = 1$ and $\mathcal{C}_{\bar{a},\bar{b}} = 1$. We also prove that $\mathcal{B}_{\bar{a},\bar{b}} = 1$. To prove this, pick any $j > i$. We have that $(a_{j+1}, b_{j+1}) = (h(a_j, g(a_j, b_j)), g(a_j, b_j))$, which reduces to $(\bar{a}, \bar{b}) = (h(\bar{a}, g(\bar{a}, \bar{b})), g(\bar{a}, \bar{b}))$. Hence, we have $\bar{b} = g(\bar{a}, \bar{b})$. Notice that since from the definition of *g*, we have that $\mathcal{B}_{\bar{a},g(\bar{a},\bar{b})} = 1$ we have that $\mathcal{B}_{\bar{a},\bar{b}} = 1$ as desired. Hence, (a^*, b^*) exists and is equal to $(\bar{a}, \bar{b})$.

To find $(\bar{a}, \bar{b})$ we enumerate the sequence $\mathcal{S}$. As established by (A28), the sequence becomes constant before *n* steps. Hence, this process is more efficient compared to the naive scheme which evaluates $\mu_{a,b}$ values for all $a, b \in [1 : n]$.

Note: In Algorithm 4 although we have defined $\mu_{a,b}$, *g*(*a*, *b*), *h*(*a*, *b*), $\mathcal{A}_{a,b}$, $\mathcal{B}_{a,b}$, and $\mathcal{C}_{a,b}$ for all $a, b \in [1 : n]$, we only require computing above for (*a*, *b*) tuples in $\mathcal{S}$.

Author Contributions M.W. and M.N. developed the idea. M.W. designed the algorithms and did the proofs. M.N. verified and provided suggestions to improve the methods. M.N supervised the work. M.W. and M.N. discussed the results and contributed to the final manuscript.

Funding Open access funding provided by SCELC, Statewide California Electronic Library Consortium. This work was supported in part by one or more of: NSF CCF-1718477, NSF SpecEES 1824418.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. Mukhopadhyay S, Sahoo S, Sinha A (2022) k-experts - online policies and fundamental limits. In: Proceedings of The 25th international conference on artificial intelligence and statistics, pp 342–365
2. Orabona F (2023) A modern introduction to online learning. arXiv:1912.13213
3. Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. Mach Learn 47:235–256
4. Lai T, Robbins H (1985) Asymptotically efficient adaptive allocation rules. Adv Appl Math 6(1):4–22. https://doi.org/10.1016/0196-8858(85)90002-8
5. Lattimore T, Szepesvári C (2020) Bandit Algorithms. Cambridge University Press, Cambridge
6. Zinkevich M (2003) Online convex programming and generalized infinitesimal gradient ascent. In: Proceedings of the twentieth international conference on machine learning. AAAI Press, ICML'03, p 928–935
7. Agarwal A, Dekel O, Xiao L (2010) Optimal algorithms for online convex optimization with multi-point bandit feedback. In: Annual conference computational learning theory
8. Hazan E, Kale S (2014) Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. J Mach Learn Res 15(1):2489–2512
9. Bubeck S, Nicolò CB (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. https://doi.org/10.1561/2200000024
10. O'Donoghue B, Lattimore T, Osband I (2021) Matrix games with bandit feedback
11. Rosenthal RW (1973) A class of games possessing pure-strategy Nash equilibria. Internat J Game Theory 2:65–67
12. Angelidakis H, Fotakis D, Lianeas T (2013) Stochastic congestion games with risk-averse players. In: Lecture Notes in Computer Science, https://doi.org/10.1007/978-3-642-41392-6_8
13. Nikolova E, Stier-Moses NE (2011) Stochastic selfish routing. In: Persiano G (ed) Algorithmic Game Theory, pp 314–325
14. Harks T, Henle M, Klimm M, et al (2022) Multi-leader congestion games with an adversary. In: Proceedings of the AAAI conference on artificial intelligence, pp 5068–5075
15. Babaioff M, Kleinberg R, Papadimitriou CH (2009) Congestion games with malicious players. Games Econ Behav 67(1):22–35
16. Syrgkanis V (2010) The complexity of equilibria in cost sharing games. pp 366–377, https://doi.org/10.1007/978-3-642-17572-5_30

17. Akkarajitsakul K, Hossain E, Niyato D et al (2011) Game theoretic approaches for multiple access in wireless networks: a survey. IEEE Commun Surv Tutor 13(3):372–395. https://doi.org/10.1109/SURV.2011.122310.000119
18. Felegyhazi M, Hubaux JP (2006) Game theory in wireless networks: A tutorial. ACM Comput Surveys
19. Garg R, Kamra A, Khurana V (2002) A game-theoretic approach towards congestion control in communication networks. SIGCOMM Comput Commun Rev 32(3):47–61
20. Aryafar E, Keshavarz-Haddad A, Wang M, et al (2013) RAT selection games in HetNets. In: 2013 Proceedings IEEE INFOCOM, pp 998–1006, https://doi.org/10.1109/INFCOM.2013.6566889
21. Felegyhazi M, Cagalj M, Bidokhti SS et al (2007) Non-cooperative multi-radio channel allocation in wireless networks. IEEE Int Conf Computer Commun 2007:1442–1450. https://doi.org/10.1109/INFCOM.2007.170
22. Li B, Qu Q, Yan Z, et al (2015) Survey on OFDMA based MAC protocols for the next generation WLAN. In: 2015 IEEE wireless communications and networking conference workshops, pp 131–135, https://doi.org/10.1109/WCNCW.2015.7122542
23. Nash J (1951) Non-cooperative games. Ann Math 54(2):286–295
24. Nash JF (1950) Equilibrium points in *n*-person games. Proc Natl Acad Sci 36(1):48–49. https://doi.org/10.1073/pnas.36.1.48
25. Aumann R (1974) Subjectivity and correlation in randomized strategies. J Math Econ 1:67–96. https://doi.org/10.1016/0304-4068(74)90037-8
26. Aumann RJ (1987) Correlated equilibrium as an expression of Bayesian rationality. Econometrica 55(1):1–18
27. Osborne MJ, Rubinstein A (1994) A Course in Game Theory, MIT Press Books, vol 1. The MIT Press
28. Cui Q, Xiong Z, Fazel M, et al (2022) Learning in congestion games with bandit feedback. ArXiv abs/2206.01880
29. Solan E, Vieille N (2002) Correlated equilibrium in stochastic games. Games Econ Behav 38(2):362–399. https://doi.org/10.1006/game.2001.0887
30. Monderer D, Shapley LS (1996) Potential games. Games Econ Behav 14(1):124–143. https://doi.org/10.1006/game.1996.0044
31. Chien S, Sinclair A (2011) Convergence to approximate Nash equilibria in congestion games. Games Econ Behav 71(2):315–327. https://doi.org/10.1016/j.geb.2009.05.004
32. Bhawalkar K, Gairing M, Roughgarden T (2010) Weighted congestion games: Price of anarchy, universal worst-case examples, and tightness. In: Lecture Notes in Computer Science, pp 17–28, https://doi.org/10.1007/978-3-642-15781-3_2
33. Milchtaich I (1996) Congestion games with player-specific payoff functions. Games Econ Behav 13(1):111–124. https://doi.org/10.1006/game.1996.0027
34. Ackermann H, Goldberg PW, Mirrokni VS et al (2008) A unified approach to congestion games and two-sided markets. Internet Math 5(4):439–458
35. Fotakis D, Kontogiannis S, Koutsoupias E et al (2009) The structure and complexity of Nash equilibria for a selfish routing game. Theor Comput Sci 410(36):3305–3326. https://doi.org/10.1016/j.tcs.2008.01.004
36. Gairing M, Lücking T, Mavronicolas M, et al (2004) Computing Nash equilibria for scheduling on restricted parallel links. In: Proceedings of the thirty-sixth annual ACM symposium on theory of computing, STOC ’04, p 613–622
37. Grundel S, Borm P, Hamers H (2013) Resource allocation games: a compromise stable extension of bankruptcy games. Math Methods Oper Res 78:149–169
38. Grundel S, Borm P, Hamers H (2018) Resource allocation problems with concave reward functions. TOP. https://doi.org/10.1007/s11750-018-0482-7
39. Thomas CD (2021) Strategic experimentation with congestion. Am Econ J Microecon 13(1):1–82
40. Bolton P, Harris C (1999) Strategic experimentation. Econometrica 67(2):349–374
41. Malanchini I, Cesana M, Gatti N (2013) Network selection and resource allocation games for wireless access networks. IEEE Trans Mob Comput 12(12):2427–2440. https://doi.org/10.1109/TMC.2012.207
42. Anshelevich E, Dasgupta A, Kleinberg J, et al (2004) The price of stability for network design with fair cost allocation. In: 45th Annual IEEE symposium on foundations of computer science, pp 295–304, https://doi.org/10.1109/FOCS.2004.68
43. Liu M, Wu Y (2008) Spectum sharing as congestion games. In: 2008 46th annual Allerton conference on communication, control, and computing, pp 1146–1153
44. Liu M, Ahmad SHA, Wu Y (2009) Congestion games with resource reuse and applications in spectrum sharing. In: 2009 International conference on game theory for networks, pp 171–179, https://doi.org/10.1109/GAMENETS.2009.5137399
45. Zhang F, Wang MM (2021) Stochastic congestion game for load balancing in mobile-edge computing. IEEE Internet Things J 8(2):778–790. https://doi.org/10.1109/JIOT.2020.3008009

46. Ibrahim M, Khawam K, Tohme S (2010) Congestion games for distributed radio access selection in broadband networks. In: 2010 IEEE global telecommunications conference 2010, pp 1–5, https://doi.org/10.1109/GLOCOM.2010.5683862
47. Le S, Wu Y, Toyoda M (2020) A congestion game framework for service chain composition in NFV with function benefit. Inf Sci 514:512–522
48. Zhang L, Gong K, Xu M (2019) Congestion control in charging stations allocation with Q-learning. Sustainability. https://doi.org/10.3390/su11143900
49. Jin C, Netrapalli P, Jordan M (2020) What is local optimality in nonconvex-nonconcave minimax optimization? In: Proceedings of the 37th international conference on machine learning, vol 119. PMLR, pp 4880–4889
50. Bertsekas D (2009) Convex optimization theory, vol 1. Athena Scientific
51. Wijewardena M, Neely MJ (2023) A two-player resource-sharing game with asymmetric information. Games 14(5):61. https://doi.org/10.3390/g14050061
52. Wijewardena M, Neely MJ (2024) Multi-player resource-sharing games with fair reward allocation. arXiv:2402.05300
53. Ibaraki T, Katoh N (1988) Resource allocation problems: algorithmic approaches. MIT Press, Cambridge