# Analyzing Queueing Problems via Bandits With Linear Reward & Nonlinear Workload Fairness

Xuchuang Wang, Hong Xie, *Member, IEEE*, and John C.S. Lui, *Fellow, IEEE*

***Abstract*—Queueing models serve as important building blocks in many networking applications such as task scheduling in mobile edge computing nodes, traffic scheduling in networks, congestion control in Internet, etc. However, queueing theory often needs to make strong assumptions about the arrival process or service rewards at each queue. In addition, fairness in serving workload among all queues is of great importance in many applications. In this paper, we address how to optimize resource allocation among multiple queues with a fairness guarantee and without any a priori knowledge of these queues' parameters. To characterize queues with unknown parameters and the fairness requirement, we formulate an online learning model with a varying and continuous action space, as well as a nonlinear utility objective. We design an online learning algorithm to tackle the problem. We prove that our algorithm has a regret upper bound of $O(\sqrt{T}\log T)$ and our model has a regret lower bound of $\Omega(\sqrt{T})$, where $T$ stands for the number of decision rounds. The asymptotic closeness of upper and lower bounds guarantees their near tightness and our algorithm's near optimality. We discuss our model's real-world applications in mobile edge computing, wireless networks, and crowdsourcing, and conduct simulations to validate our algorithm's effectiveness.**

***Index Terms*—Fairness, linear bandits, queueing problems.**

## I. Introduction

Queueing models serve as important building blocks in many networking applications, e.g., in mobile edge computing, it is applied to schedule offloaded tasks to servers [1], [2], [3], [4]. In 5G wireless networks, it is applied to schedule traffic flows [5], [6], [7], [8]. In mobile crowdsourcing applications, it is applied to design task allocation policies [9], [10], [11], [12]. Essentially, queueing models enable one to study resource allocation problems and devise various allocation policies. To motivate our study, consider the following toy example in Table I composed of two queues $Q_a, Q_b$ and a single server $S$, where $\mathcal{B}(\cdot)$ stands for Bernoulli random variables (r.v.s) [13, § 2.2].

TABLE I
A Two-Queue Toy Example

| | Service reward | Initial queue length | Interarrival r.v. |
|---|---|---|---|
| $Q_a$ | 2 | 4 | $\mathcal{B}(0.1)$ |
| $Q_b$ | 1 | 3 | $2\cdot\mathcal{B}(0.9)$ |

In the next time slot, the single server $S$ can serve one unit of queue length (job) of one queue. By allocating the server $S$ to $Q_a$, one can earn a reward of 2 which is higher than the reward of 1 from allocating the server to $Q_b$. To maximize the total reward, the one needs to serve $Q_a$ whenever it is non-empty. In this case, queue $Q_b$'s job would accumulate as time elapses and its average queue length would be much longer than $Q_a$'s, which implies that the above policy is *unfair*. Since fairness is of great importance in many applications, simply maximizing the total reward may not be a suitable objective. For example, in network bandwidth scheduling, although serving some flows (e.g., video streams) can please some users, other flows (e.g., large file transfer) should be served fairly also.

The queue length of a flow represents the amount of backlogged data, and it can affect the quality of experience (QoE) [14] for its corresponding users. To measure "fairness" of balancing the QoE among queues, the difference of queue lengths can be used as a metric. In the example of Table I, after serving one job of $Q_a$, the expected queue difference is $|(4\underline{-1}+0.1)-(3+2\times 0.9)| = 1.7$, while if serving $Q_b$ instead, the queue difference would be $|(4+0.1)-(3\underline{-1}+2\times 0.9)| = 0.3$. Thus, in terms of fairness, the server should be allocated to $Q_b$ instead of $Q_a$. This shows that a fair allocation may not be simply serving the longest queue $Q_a$. This counterintuitive observation illustrates that to maintain fairness in queueing models, one should consider not only the current queue lengths, but also the job interarrival random variables [15, § 2.2].

From the above example, to balance reward and fairness, one needs to know both the arrival processes of these queues and the rewards of serving them. However, such information is often unavailable in many real world applications. Take network packet scheduling as an example. Since the network performance depends on the number of flows and their routing paths (both of which change frequently), it is difficult to know the service rewards and arrival rates of each flow in advance.

In this paper, we develop an online learning framework to address the above challenge. In particular, to tackle the

Manuscript received 19 February 2022; revised 22 December 2022; accepted 9 May 2023. Date of publication 15 May 2023; date of current version 6 March 2024. The work of John C.S. Lui and Xuchuang Wang were supported in part by the RGC's GRF under Grant 14215722. The work of Hong Xie was supported by Chongqing Talents: Exceptional Young Talents Project under Grant cstc2021ycjhbgzxm0195, the Chinese Academy of Sciences "Light of West China" Program, the Key Cooperation Project of Chongqing Municipal Education Commission under Grants HZ2021008 and HZ2021017, and the "Fertilizer Robot" project of Chongqing Committee on Agriculture and Rural Affairs. Recommended for acceptance by D. Nguyen. *(Corresponding author: Hong Xie.)*

Xuchuang Wang and John C.S. Lui are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: xcwang@cse.cuhk.edu.hk; cslui@cse.cuhk.edu.hk).

Hong Xie is with the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, Sichuan 400065, China (e-mail: xiehong2018@foxmail.com).

Digital Object Identifier 10.1109/TMC.2023.3276158

unknown environment and to satisfy the fairness requirement, we propose the QLBF (*Queueing Linear Bandits with nonlinear Fairness guarantee*) online learning model. We design an upper confidence based online learning algorithm, called QLBF-UCB (*Upper Confidence Bound*) to sequentially allocate the resource among multiple queues.

Note that the QLBF model we study is a *more general setting* than our motivating example. The QLBF model consists of one resource allocator (server) with *limited but arbitrarily dividable* resources, and multiple queues whose service rewards and interarrival distributions are unknown *a priori*. Different from classic queueing theory's interests, like customer waiting time and server idle time [16], QLBF focuses on online decision making. In each time slot, the server assigns its resources to these queues and then, after a deterministic service time, observes the total service reward and the new arrivals of each queue. In this model, both the action space and the new arrivals are *"continuous"*. Besides a linear structure reward, we employ a *nonlinear* variance function of these queue lengths to measure QLBF's fairness which can be mapped to many fairness metrics in queueing applications. Combining the fairness and service rewards, we define a *"nonlinear utility"* as QLBF's objective. Our online learning task aims to to maximize its total service reward as well as maintain fairness among multiple queues. We refer to Section VI for a detailed explanation of QLBF's real-world application scenarios, including bandwidth scheduling of 5G wireless networks, resources scheduling of mobile-edge computing nodes, and task allocation in mobile crowdsourcing.

More general than traditional queueing systems, the QLBF model takes service rewards into consideration. The continuous action space and state space further complicate the model, making it difficult, if not impossible, to find an optimal policy under its setting even when the model parameters are given. We therefore choose a utility greedy policy as our learning benchmark when designing algorithms. Extensive numerical comparisons with other policies such as longest queue first, shortest queue first, and round-robin validate the greedy policy's effectiveness. Our work focuses on studying the learnability of the QLBF model and designing online learning algorithms to address it.

Our QLBF-UCB algorithm *extends* the upper confidence bound (UCB) algorithm [17] to address our new online learning task. The algorithm's UCB consists of two terms: the linear structure reward's and the nonlinear fairness term's. We adapt linear bandits' techniques [18] to estimate the linear structure reward's UCB. Designing the nonlinear fairness term's UCB is more challenging. Because it depends on the unknown and nonparametric arrival processes of multiple queues. To address the issue, we employ the *Dvoretzky-Kiefer-Wolfowitz inequality* [19, Theorme 7.1] to construct the confidence band of the arrivals' cumulative distribution functions. We show that the QLBF-UCB algorithm has an $O(\sqrt{T}\log T)$ regret upper bound and the QLBF model has a $\Omega(\sqrt{T})$ regret lower bound, where $T$ stands for the decision time horizon. The fact that there is only a logarithmic factor missing in the lower bound implies that QLBF-UCB is nearly optimal and our proposed bounds are nearly tight. We conduct simulations to show QLBF-UCB's learning effectiveness in the QLBF model. The results validate our sublinear regret bound.

TABLE II
NOTATION

| | |
|---|---|
| Space & Sets: | |
| $\mathcal{K}$ | queue index set |
| $\mathcal{A}_t$ | feasible allocation space |
| $\mathcal{H}_t$ | action and observation history set (sequence) |
| $\mathcal{C}_t$ | reward estimate confidence set |
| $\mathcal{F}_t$ | arrival CDF estimate's confidence function space |
| Functions: | |
| $R_t$ | linear reward function $\mathcal{A}_t \to \mathbb{R}$ |
| $V$ | variance function $\mathbb{R}^k \to \mathbb{R}$ |
| $\bar{V}_t$ | fairness function $\mathbb{R}^k \to \mathbb{R}$ |
| $u_t$ | utility function $\mathcal{A}_t \to \mathbb{R}$ |
| $\pi$ | allocation policy History Space $\to \mathcal{A}_t$ |
| $\text{UCB}_t^R$ | upper confidence bound of reward $\mathcal{A}_t \to \mathbb{R}$ |
| $\text{LCB}_t^{\text{fair}}$ | lower confidence bound of fairness $\mathcal{A}_t \to \mathbb{R}$ |
| $\text{UCB}_t$ | upper confidence bound of utility $\mathcal{A}_t \to \mathbb{R}$ |
| $F^{(k)}$ | CDF of queue $k$'s arrival distribution |
| $F$ | joint CDF of all queues' arrival distribution |
| Vectors & Matirces: | |
| $\boldsymbol{\mu}^*$ | per-unit service reward $(\mu_k^* : k \in \mathcal{K})$ |
| $\boldsymbol{w}_t$ | queue priority weight vector $(w_{k,t} : k \in \mathcal{K})$ |
| $\boldsymbol{\Lambda}_t$ | request arrival vector $(\Lambda_{k,t} : k \in \mathcal{K})$ |
| $\boldsymbol{L}_t$ | queue length vector $(L_{k,t} : k \in \mathcal{K})$ |
| $\boldsymbol{A}_t$ | allocation vector $(A_{k,t} : k \in \mathcal{K})$ |
| $\boldsymbol{A}_t^*$ | utility greedy allocation vector |
| $\boldsymbol{W}_t$ | auxiliary matrix for least square estimate of reward |
| Scalars: | |
| $k$ | queue index |
| $K$ | queue number |
| $t$ | time index |
| $T$ | time horizon |
| $c$ | server capacity |
| $d$ | 2-norm bound of reward |
| $s$ | coefficient before fairness term in utility function |
| $\xi$ | coefficient of regularizer in fairness metric |
| $\lambda$ | coefficient of regularizer in estimating reward |
| $\delta$ | confidence parameter |
| $\eta_t$ | subgaussian zero-mean noise |
| $\sqrt{\beta_t}$ | confidence range of set $\mathcal{C}_t$ |
| $\sqrt{\gamma_t}$ | confidence range of set $\mathcal{F}_t$ |
| $\text{reg}_t$ | instantaneous regret |
| $\text{Reg}_T$ | expected regret |
| $\text{Reg}_T^*$ | minimax regret |

The paper organizes as follows. Section II presents the related work. Section III gives the detail of our QLBF model and formulates the online learning problem. In Section IV, we devise our QLBF-UCB algorithm and derive its UCB formula. Then, we present QLBF-UCB theoretical regret bounds in Section V. In Section VI, we discuss potential applications of our QLBF model in 5G networks, mobile-edge computing systems, and crowdsourcing. In Section VII, we provide simulations to validate the efficacy of offline utility greedy policy and corroborate our QLBF-UCB algorithm's theoretical analysis results. Our theoretical results' rigorous proofs are presented in Section VIII. At last, we conclude our paper in Section IX.

## II. Related Works

While there is a line of works applying restless bandits model (with known parameters, i.e., offline) to study queueing based scheduling tasks (e.g., see survey [20]), only few works considered queueing model in the online learning context [21], [22]. Krishnasamy et.al. [21] studied the case that an operator allocates jobs in a queue to multiple servers, where the servers with different service rates are modeled as arms with different reward means and the allocation of jobs in the single queue decided by the operator. Stahlbuhk et.al. [22] proposed an algorithm for this setting achieving constant regret when the arrival rate of jobs is less then the optimal server (arm)'s service rate. Our QLBF model is different from their settings in two ways: (1) the QLBF model considers heterogeneous service rewards among multiple queues as *arms* while their model only contained one single queue as an *operator*; (2) QLBF's action set is continuous and varying while previous works' are fixed, finite, and discrete.

Recently, fairness is of interest in sequential learning (e.g., MAB) [23], [24], [25], [26]. Joseph et.al. [23] first considered fairness in classical and contextual bandits and provided algorithms to achieve *individual* fairness in both setting. Li et.al. [24] studied the sleeping semi-combinatorial bandits with fairness, where, to guarantee the fairness, they required that the pull fractions of arms should be greater than a pre-specified vector. Following Li et.al. [24], Patil et.al. [27] proposed a fairness-aware regret taking this fairness constraint into consider; Chen et.al. [28] introduced this fairness constraint to the contextual MAB model; [29], [30] studied this constraint in Federated MAB model, etc. Bistritz et.al. [26] introduced the max-min fairness into distributed multi-player bandits. Different from their fairness measures, our QLBF model is the *first to use the variance of queue lengths as fairness metric*, which can be easily extended to classical Jain's fairness index [31] and the newly proposed quality of experience (QoE) fairness [32]. We also note that variance was also included in the utility function of Risk-averse multi-armed bandits (e.g., [33], [34]). Our utility function is deferent from theirs as our variance (fairness) is of queue lengths, while their variance only depended on rewards. Wang et al. [35] introduced the fairness-of-exposure constraints to MAB and linear bandits to avoid winner-takes-all allocation. Their objective focused on maximizing the merit of fairness which is different from our utility function taking both reward and fairness into account.

UCB algorithms had been extensively studied both in general MAB model [36], [37], [38] and in linear structured MAB model [18], [39], [40]. Peter et.al. [38] were the first to achieve the uniform sublinear regret bound in MAB via the UCB algorithm. Abbasi-Yadkori et.al. [18] proposed a linear UCB algorithm for linear structure MAB and a uniform UCB algorithm for general MAB model. Although our model contains a linear reward term as linear bandit did, the nonlinear fairness term in our utility makes these linear bandit algorithm not applicable. Using queue lengths as a varying environment shares a similarity with MAB's contextual setting (e.g., [41], [42], [43], [44]). Different from their settings, our utility function directly converts the queue lengths to the fairness term instead of taking them as side information.

## III. Model and Problem Formulation

In this section, we step by step formulate the QLBF model: first the queueing model, then the action and feedback, and finally the fairness and utility function. Lastly, we define our online learning problem and its performance criterion.

### A. Queueing Model

Consider a finite number of $K \in \mathbb{N}_+$ queues. Each queue is associated with one type of requests or tasks. We use a discrete time system indexed by $t \in \mathbb{N}_+$ to characterize these queues' arrival processes. Let $\Lambda_{k,t} \in \mathbb{R}_+$ denote the amount of requests arriving to queue $k \in \mathcal{K} \triangleq \{1, 2, \ldots, K\}$ at time $t$. Here, $\Lambda_{k,t}$ is a random variable with a bounded support $[0, b]$, where $b \in \mathbb{R}_+$. The range of $\Lambda_{k,t}$ is allowed to be discrete (e.g., modeling number of arrivals) or continuous (e.g., modeling fluid queue). For each given queue $k$, the arrivals of $\Lambda_{k,t}$ across $t$ are independent and identically distributed. Also, $\Lambda_{k,t}$ across $k$ are independent. Let $L_{k,t} \geq 0$ denote the length of the $k$-th queue in time slot $t$. For simplicity, define $\boldsymbol{L}_t \triangleq (L_{k,t} : k \in \mathcal{K})$, $\boldsymbol{\Lambda}_t \triangleq (\Lambda_{k,t} : k \in \mathcal{K})$ and denote $\mathcal{D}$ as the distribution that the random vector $\boldsymbol{\Lambda}_t$ follows.

Let $c \in \mathbb{R}_+$ denote the server's total amount of capacity (or resources) which is arbitrarily dividable. The decision maker needs to sequentially allocates the capacity to serve queues in each time slot $t$. We consider a deterministic service time setting, i.e., each service would finish at the end of each time slot. After one time slot's service, each queue length's reduction is equal to its allocated capacity.

### B. Action and Feedback

Let $A_{k,t} \in [0, c]$ denote the amount of capacity allocated to queue $k$ in time slot $t$ and $\boldsymbol{A}_t \triangleq (A_{k,t} \in [0, c] : k \in \mathcal{K})$ denote the allocation vector.[1] The total allocated capacities in one time slot is no greater than $c$, i.e., $\|\boldsymbol{A}_t\|_1 \leq c$. As the queue length represents the demand of its corresponding task, the assigned capacity to each queue should not exceed the length in its preceding time slot, i.e., $A_{k,t} \leq L_{k,t-1}$ for any $k \in \mathcal{K}$. Thus, the space of all the feasible capacity allocations in time slot $t$ is

$$\mathcal{A}_t \triangleq \{\boldsymbol{A} : \boldsymbol{0} \leq \boldsymbol{A} \leq \boldsymbol{L}_{t-1}, \|\boldsymbol{A}\|_1 \leq c\}.$$

We consider a linear reward function associated with each action $\boldsymbol{A}_t \in \mathcal{A}_t$ as follows:

$$R_t(\boldsymbol{A}_t) \triangleq \boldsymbol{A}_t^T \boldsymbol{\mu}^* + \eta_t, \tag{1}$$

where $\boldsymbol{\mu}^* \triangleq (\mu_k^* : k \in \mathcal{K}) \in \mathbb{R}_+^K$ and $\eta_t \in \mathbb{R}$ is a subgaussian zero-mean noise [45, § 2.3]. Here, $\mu_k^*$ can be interpreted as the average per-unit service reward of queue $k$ and it is *unknown* to the decision maker. Also, all $\mu_k^*$ are bounded, i.e., exists $d \in \mathbb{R}_+$ such that $\|\boldsymbol{\mu}^*\|_2 \leq d$.

In time slot $t$, after allocating capacities according to $\boldsymbol{A}_t$, the total reward $R_t$ and the number of new arrivals $\boldsymbol{\Lambda}_t$ will be revealed to the decision maker. The queue length $\boldsymbol{L}_t$ is the total arrivals up to time $t$ minus the cumulative allocated capacities up

[1] Throughout this paper, we use boldface notations to represent $K$-dimension vectors whose entries correspond to all $K$ queues accordingly. When a notation defined in boldface (e.g., $\boldsymbol{A}_t$), is used in regular font (e.g., $A_{k,t}$), it stands for an entry of this vector corresponding to a specific queue.

to time $t$. So, the queue length $\boldsymbol{L}_t$ can be calculated as follows:

$$\boldsymbol{L}_t = \sum_{l=1}^{t}(\boldsymbol{\Lambda}_l - \boldsymbol{A}_l) = \boldsymbol{L}_{t-1} + \boldsymbol{\Lambda}_t - \boldsymbol{A}_t. \tag{2}$$

### C. Fairness and Utility Function

We measure the fairness via the variance of queue lengths. Let $V : \mathbb{R}^K \to \mathbb{R}_+$ denote the variance function, which is

$$V(\boldsymbol{L}_t) \triangleq \frac{1}{K}\sum_{k=1}^{K}\left(L_{k,t} - \frac{1}{K}\sum_{i=1}^{K} L_{i,t}\right)^2.$$

*Note that our variance fairness definition can be easily extended to many other fairness metrics* via multiplying a constant factor, taking its reciprocal or square root. For example, the coefficient of variation $V(\boldsymbol{L}_t)/(\sum_{i=1}^{K} L_{i,t}/K)$ and its inverse are classical fairness measures in network [31], [46], and the recently proposed *quality of experience* (QoE) fairness [32], [47] is one minus the normalized standard variance, i.e., $1 - \sqrt{V(\boldsymbol{L}_t)/\max_{\boldsymbol{L}'} V(\boldsymbol{L}')}$. For another thing, the min-max and proportional fairness metrics [48, § III] are not suitable for our model. Because both metrics are fairness conditions—one allocation is either max-min (proportional) fair or not. Since the min-max and proportional fairness metrics are not quantitative expressions for queueing systems, both cannot be used in our quantitative utility function.

We note that the variance function $V(\boldsymbol{L}_t)$ is equal to $V(\boldsymbol{L}_t + \delta)$ where each queue length varies the same value $\delta$, and the smaller length vector is preferable. To differentiate between $V(\boldsymbol{L}_t)$ and $V(\boldsymbol{L}_t + \delta)$ in fairness measurement and motivate the algorithm to minimize the queue length vector, we add a 2-norm regularization term $\|\boldsymbol{L}_t\|_2^2$ to the variance function. Let $\boldsymbol{w}_t \triangleq (w_{k,t} : k \in \mathcal{K})$ denote a time-varying weight which captures the varying priorities among queues. These weights depend on the instantaneous environment and may not be observed until the $t$th allocation. So, the fairness associated with weighted queue length vector $\boldsymbol{w}_t \circ \boldsymbol{L}_t$ is defined as

$$V(\boldsymbol{w}_t \circ \boldsymbol{L}_t) + \xi \|\boldsymbol{w}_t \circ \boldsymbol{L}_t\|_2^2,$$

where the $\circ$ operator stands for element-wise (Hadamard) product and $\xi \in \mathbb{R}^+$ scales the regularizer.

This weight $\boldsymbol{w}_t$ allows our fairness metric to balance the service of queues in various ways, including balancing queue delays. For example, (1) if the $w_{k,t}$ are the same for all $k$, our fairness aims to uniformize all queues' length; (2) if the $w_{k,t}$ is equal to the reciprocal of the arrival rate for each queue (e.g., assume the arrival rate is known or estimated with high accuracy), our fairness is aim to balance all queues' delay when the system is stable (cf. Little's Law [16, Theorem 1]). In a word, our fairness function is versatile and, with well selected weights, can be used to balance the delay of different queues.

As the queue length $\boldsymbol{L}_t$ is equal to $\boldsymbol{L}_{t-1} + \boldsymbol{\Lambda}_t - \boldsymbol{A}_t$ and the new arrivals $\boldsymbol{\Lambda}_t$ is unknown when making decisions, we need to calculate the variation's expectation with respect to the interarrival random vector $\boldsymbol{\Lambda}_t$. Therefore, we rewrite the variation as $\bar{V}_t(\boldsymbol{\Lambda}_t; \boldsymbol{A}_t, \boldsymbol{L}_{t-1}, \boldsymbol{w}_t)$ which is equal to

$$V(\boldsymbol{w}_t \circ (\boldsymbol{L}_{t-1} + \boldsymbol{\Lambda}_t - \boldsymbol{A}_t)) + \xi \|\boldsymbol{w}_t \circ (\boldsymbol{L}_{t-1} + \boldsymbol{\Lambda}_t - \boldsymbol{A}_t)\|_2^2.$$

We then take the expectation of the variation $\mathbb{E}_{\boldsymbol{\Lambda}_t \sim \mathcal{D}}[\bar{V}_t(\boldsymbol{\Lambda}_t)]$ as a penalty for measuring fairness. Finally, we define the utility function $u$ as follows

$$\begin{aligned} u_t(\boldsymbol{A}_t) &\triangleq \mathbb{E}[R_t(\boldsymbol{A}_t)] - s \cdot \mathbb{E}_{\boldsymbol{\Lambda}_t \sim \mathcal{D}}\left[\bar{V}_t(\boldsymbol{\Lambda}_t; \boldsymbol{A}_t, \boldsymbol{L}_{t-1}, \boldsymbol{w}_t)\right], \\ &= \boldsymbol{A}_t^T \boldsymbol{\mu}^* - s \cdot \mathbb{E}_{\boldsymbol{\Lambda}_t \sim \mathcal{D}}\left[\bar{V}_t(\boldsymbol{\Lambda}_t)\right], \end{aligned} \tag{3}$$

where $s \in \mathbb{R}_+$ is a positive factor for controlling the relative scale between both terms.

### D. Online Learning Problem

The decision maker aims to maximize the total expected utility in a finite time horizon $T$. Because the QLBF's optimal policy is difficult to find (and may not even exist due to the time-varying weight $\boldsymbol{w}_t$) and our work focuses on the learnability of the QLBF model, we take the one-step utility greedy action with full information as our learning benchmark. The utility greedy action in time slot $t$ can be expressed as

$$\boldsymbol{A}_t^* \in \arg\max_{\boldsymbol{A} \in \mathcal{A}_t} u_t(\boldsymbol{A}). \tag{4}$$

We extensively validate the policy's effectiveness in Section VI-I-A.

Denote the actions and observations history up to time $t$ as

$$\mathcal{H}_t \triangleq ((\boldsymbol{A}_1, R_1, \boldsymbol{\Lambda}_1), (\boldsymbol{A}_2, R_2, \boldsymbol{\Lambda}_2), \dots, (\boldsymbol{A}_t, R_t, \boldsymbol{\Lambda}_t)).$$

Then, the online learning problem is to design a policy

$$\pi : \mathcal{H}_{t-1} \mapsto \boldsymbol{A}_t \in \mathcal{A}_t$$

for any time $t$, such that the accumulated utility under those actions in the algorithm is as large as possible.

To measure a learning algorithm's performance, we define the *regret*, which is the accumulative difference between the learning algorithm's action $\boldsymbol{A}_t = \pi(\mathcal{H}_{t-1})$'s utility and the given benchmark action $\boldsymbol{A}_t^*$'s in each time slots as follows:

$$\text{Reg}_T \triangleq \sum_{t=1}^{T}\left(u_t(\boldsymbol{A}_t^*) - u_t(\boldsymbol{A}_t)\right). \tag{5}$$

To minimize the regret $\text{Reg}_T$, we design the QLBF-UCB algorithm in Section IV and derive its theoretical performance guarantee in Section V.

## IV. Algorithm Design

In this section, we first present the main idea of QLBF-UCB, which is to allocate resources according to the optimistic estimate of utility. Then, we derive the explicitly formulas to calculate the utility function's UCB.

### A. The Main Idea of QLBF-UCB

Our QLBF-UCB algorithm shares the same principle as Lin-UCB [18]: *optimism in the face of uncertainty*. Take the linear reward term $\boldsymbol{A}^T\boldsymbol{\mu}^*$ as an example, where $\boldsymbol{A} \in \mathcal{A}_t$. Suppose we can derive a *confidence set* $\mathcal{C}_t$ for reward mean vector $\boldsymbol{\mu}^*$ from history $\mathcal{H}_{t-1}$ such that $\boldsymbol{\mu}^*$ is in the set with high probability. Then, with the high probability, the $\max_{\boldsymbol{\mu}\in\mathcal{C}_t} \boldsymbol{A}^T\boldsymbol{\mu}$ is no less than $\boldsymbol{A}^T\boldsymbol{\mu}^*$ for any action $\boldsymbol{A}$ in the feasible action space $\mathcal{A}_t$, and it is named as upper confidence bound (UCB) of reward $\boldsymbol{A}^T\boldsymbol{\mu}^*$. The $\max_{\boldsymbol{\mu}\in\mathcal{C}_t} \boldsymbol{A}^T\boldsymbol{\mu}$ is an *optimistic* estimate of the linear reward term corresponding to action $\boldsymbol{A}$. For simplicity, denote

$$\mathrm{UCB}_t^R(\boldsymbol{A}) \triangleq \max_{\boldsymbol{\mu}\in\mathcal{C}_t} \boldsymbol{A}^T\boldsymbol{\mu}.$$

Similarly, after deriving a confidence space for the new arrival distribution $\mathcal{D}$ (specifically, we later derive its cumulative distribution function (CDF) $F$'s confidence function space $\mathcal{F}_t$), the optimism estimate of the *penalty* fairness term is minimizing within the confidence space $\mathcal{F}_t$, i.e., its lower confidence bound (LCB), as follows

$$\mathrm{LCB}_t^{\mathrm{fair}}(\boldsymbol{A}) \triangleq \min_{F'\in\mathcal{F}_t} \mathbb{E}_{\boldsymbol{\Lambda}_t\sim F'}\left[\bar{V}_t(\boldsymbol{\Lambda}_t)\right].$$

We defer the derivations of $\mathrm{UCB}_t^R$ and $\mathrm{LCB}_t^{\mathrm{fair}}$ to the next subsection. Here, we assume that they are given and use them to design our QLBF-UCB algorithm. The utility function has the following upper bound with high probability,

$$\begin{aligned} u_t(\boldsymbol{A}) &\leq \max_{\boldsymbol{\mu}\in\mathcal{C}_t} \boldsymbol{A}^T\boldsymbol{\mu} - s\cdot \min_{F'\in\mathcal{F}_t} \mathbb{E}_{\boldsymbol{\Lambda}_t\sim F'}\left[\bar{V}_t(\boldsymbol{\Lambda}_t)\right] \\ &\leq \mathrm{UCB}_t^R(\boldsymbol{A}) - s\cdot \mathrm{LCB}_t^{\mathrm{fair}}(\boldsymbol{A}) \\ &\triangleq \mathrm{LCB}_t^{\mathrm{fair}}(\boldsymbol{A}). \end{aligned} \tag{6}$$

To learn the utility greedy policy defined in (4) and reduce the regret of this policy, our algorithm greedily selects the action which maximizes $\mathrm{UCB}_t(\boldsymbol{A})$. Namely, in time slot $t$, the algorithm chooses an action from $\mathrm{argmax}_{\boldsymbol{A}\in\mathcal{A}_t}\mathrm{UCB}_t(\boldsymbol{A})$. The optimism in the face of uncertainty principle implies that our algorithm design would have a good performance: Although the performance of the QLBF-UCB algorithm does not always improve in each time slot, the principle guarantess that the algorithm converges to the one-step greedy policy in a fast speed. Its rigorous analysis is presented in Section V.

We provide QLBF-UCB's pseudo-code in Algorithm 1. The remaining challenge is to derive expressions for $\mathrm{LCB}_t^{\mathrm{fair}}(\boldsymbol{A})$ and $\mathrm{UCB}_t^R(\boldsymbol{A})$. We present both terms' derivations in the rest of this section. Although the high-level idea of QLBF-UCB algorithm is the same as the LinUCB, which is the well-known optimism in the face of uncertainty principle in online learning, introducing the nonlinear term into the objective brings new challenges that are unique in QLBF-UCB. For one thing, we need to derive a lower confidence bound for the nonlinear fairness term which requires new techniques, e.g., utilizing the *Dvoretzky-Kiefer-Wolfowitz inequality*. For another thing, after deriving this lower bound, we also need to choose a suitable ratio factor (see the $\sqrt{\gamma}$ in (7)) in the QLBF-UCB algorithm to make sure that the algorithm can enjoy a sublinear regret upper bound.

We note that there is a line of work studying the bandit convex optimization (BCO) initiated by Flaxman et al.[49]. For example, Hazan and Levy [50] attained the $O(\sqrt{T})$ regret bound under the assumptions of strong convexity and smoothness; Suggala et al. [51] replaced the strong convex loss assumption by quadratic loss assumption while still achieving the optimal $O(\sqrt{T})$ regret; Hazan and Levy [52] relaxed the strong convexity assumption for all $T$ time slots to the strong convexity for a part of time slots (e.g., $T^{3/4}$) while still recovering the $O(\sqrt{T})$ bound. Although BCO's function assumption covers our nonlinear (quadratic utility) case, BCO's learning setting is adversarial, i.e., the reward functions change adversarially in different time slots, while in our stochastic setting, the quadratic (utility) function is fixed with an additive stochastic noise. Therefore, these algorithms of BCO was not applicable to our setting.

**Algorithm 1:** The QLBF-UCB Algorithm.

```
Input: $\lambda \in \mathbb{R}_+, K, T \in \mathbb{N}_+$.
Initialize time index $t = 1$, confidence sets $\mathcal{C}_t, \mathcal{F}_t$ with
    their largest set radii (see (7) and (11)), and the initial
    queue length $\boldsymbol{L}_{t-1}$ as an all one vector.
while $t \leq T$ do
    Update the $\mathrm{UCB}_t(\cdot)$ by $\mathcal{C}_t, \mathcal{F}_t, \boldsymbol{L}_{t-1}$ via (6), (9) and
    (12).
    Select action $\boldsymbol{A}_t \in \mathrm{argmax}_{\boldsymbol{A}\in\mathcal{A}_t}\mathrm{UCB}_t(\boldsymbol{A})$.
    Observe the linear reward $R_t$ and new arrivals $\boldsymbol{\Lambda}_t$.
    $\mathcal{H}_t \leftarrow \mathcal{H}_{t-1} \cup \{(\boldsymbol{A}_t, R_t, \boldsymbol{\Lambda}_t)\}$.
    Update confidence set $\mathcal{C}_{t+1}, \mathcal{F}_{t+1}, \boldsymbol{L}_t$ from history $\mathcal{H}_t$
    via (10), (8), and (2) respectively.
    $t \leftarrow t + 1$.
end while
```

## B. Deriving UCB Expressions

*Deriving $LCB_t^{fair}(\boldsymbol{A})$:* We first estimate the CDF of arrival distribution $\mathcal{D}$ from historical arrival vectors $(\boldsymbol{\Lambda}_s)_{s=1}^{t-1}$. Denote the CDF of queue $k$'s arrival distribution as $F^{(k)}$ and all queues' joint CDF as

$$F \triangleq \prod_{k=1}^{K} F^{(k)},$$

which is due to queues' independence. We separately estimate each queue $k$'s CDF via its empirical mean:

$$\hat{F}_{t-1}^{(k)}(\Lambda) \triangleq \frac{\sum_{s=1}^{t-1}\mathbb{1}\{\Lambda_{k,s}\leq\Lambda\}}{t-1}, \quad \forall\Lambda\in[0,b],$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. Then, the joint CDF $\hat{F}_{t-1}$ is equal to the multiplication of each queue's empirical CDF, i.e., $\hat{F}_{t-1} = \prod_{i=1}^{K}\hat{F}_{t-1}^{(k)}$.

In the next lemma, we construct the confidence band for estimated CDF $\hat{F}_t^{(k)}$ which is based on the *Dvoretzky–Kiefer–Wolfowitz inequality*. Please refer to Section VIII for detail proofs of our propositions, lemmas, and theorems.

*Lemma 1:* Denote a confidence band (function space) as

$$\mathcal{F}_t^{(k)} \triangleq \left\{ f\in C_b([0,b]) : \hat{L}_t^{(k)}(\Lambda) \leq f(\Lambda) \leq \hat{U}_t^{(k)}(\Lambda), \forall\Lambda\in[0,b] \right\}$$

where the lower band function $\hat{L}_t^{(k)}$ and $\hat{U}_t^{(k)}$ are defined as the lower and upper band function respectively

$$\hat{L}_t^{(k)} \triangleq \max\left\{\hat{F}_t^{(k)} - \sqrt{\gamma_t}, 0\right\},\ \hat{U}_t^{(k)} \triangleq \min\left\{\hat{F}_t^{(k)} + \sqrt{\gamma_t}, 1\right\},$$

and the factor $\sqrt{\gamma_t}$ is defined as

$$\sqrt{\gamma_t} \triangleq \sqrt{\frac{1}{2t}\log\left(2K(t+T)^2\right)}. \tag{7}$$

Then, the probability that the true CDF $F^{(k)}$ of queue $k$ is in space $\mathcal{F}_t^{(k)}$ satisfies

$$\mathbb{P}\left(F^{(k)} \in \mathcal{F}_t^{(k)}, \forall t \leq T\right) \geq 1 - \frac{1}{2KT}.$$

As Lemma 1 holds for each queue and these queues' arrival $\Lambda_{k,t}$ (across $k$) are independent, the joint CDF $F$ is in the function space

$$\mathcal{F}_t \triangleq \Bigg\{ f \in C_b\left([0,b]^K\right) : \forall \mathbf{\Lambda} \in [0,b]^K, \times \prod_{k=1}^{K} \hat{L}_t^{(k)}(\Lambda_k) \leq f(\mathbf{\Lambda}) \leq \prod_{k=1}^{K} \hat{U}_t^{(k)}(\Lambda_k) \Bigg\} \tag{8}$$

with probability $1 - 1/2\,T$.

With the confidence bands constructed in Lemma 1, we derive the lower confidence bound of the fairness term in (9). We defer its detailed derivation to Section VIII-A.

$$\begin{aligned}
&\min_{F\in\mathcal{F}_t} \mathbb{E}_{\mathbf{\Lambda}_t\sim F}\bar{V}_t(\mathbf{\Lambda}_t) \geq \bar{V}_t(b\cdot\mathbf{1})\\
&+\sum_{1\leq i<j\leq K}\int_0^b\int_0^b \hat{U}_{t-1}^{(i)}(\Lambda_{i,t})\hat{U}_{t-1}^{(j)}(\Lambda_{j,t})\\
&\times\frac{\partial^2\bar{V}_t(b,\ldots,b,\Lambda_{i,t},b,\ldots,b)}{\partial\Lambda_{j,t}\partial\Lambda_{i,t}}d\Lambda_{i,t}d\Lambda_{j,t}\\
&-\sum_{i=1}^{K}\int_0^{q_{i,t}}\hat{L}_{t-1}^{(k)}(\Lambda_{i,t})\frac{\partial\bar{V}_t(b,\ldots,b,\Lambda_{i,t},b,\ldots,b)}{\partial\Lambda_{i,t}}d\Lambda_{i,t}\\
&-\sum_{i=1}^{K}\int_{q_{i,t}}^{b}\hat{U}_{t-1}^{(k)}(\Lambda_{i,t})\frac{\partial\bar{V}_t(b,\ldots,b,\Lambda_{i,t},b,\ldots,b)}{\partial\Lambda_{i,t}}d\Lambda_{i,t},
\end{aligned} \tag{9}$$

where $q_{k,t} \triangleq b\mathbb{1}_{\{z_{k,t}\geq b\}} + z_{k,t}\mathbb{1}_{\{0<z_{k,t}<b\}}$ and $z_{k,t}$ is the unique zero of $\bar{V}_t$'s partial derivative with respect to $\Lambda_{k,t}$.

For practicality, we use the RHS of (9) as the closed-form formula of the fairness term's lower confidence bound, i.e., $\mathrm{LCB}_t^{\mathrm{fair}}(\boldsymbol{A}_t)$. The reasons are that (1) the original $\mathrm{LCB}_t^{\mathrm{fair}}(\boldsymbol{A})$ is defined as the result of the optimization $\min_{F\in\mathcal{F}_t}\mathbb{E}_{\mathbf{\Lambda}_t\sim F}\bar{V}_t(\mathbf{\Lambda}_t)$ and it can be computationally expensive or even intractable, and (2) the lower bound's gap, depending on the space $\mathcal{F}_t$'s radius $\sqrt{\gamma_t}$, would shirk to zero as historical data accumulates so that the RHS of (9) is close to the value of $\min_{F\in\mathcal{F}_t}\mathbb{E}_{\mathbf{\Lambda}_t\sim F}\bar{V}_t(\mathbf{\Lambda}_t)$. For the same reasons, we also abuse the notation $\mathrm{UCB}_t^R$ as its upper bound next.

*Deriving $UCB_t^R(\boldsymbol{A})$:* With all previous rewards $\{R_l\}_{l=1}^t$, we apply the regularized least square estimator to estimate $\hat{\mu}_t$:

$$\min_{\boldsymbol{\mu}} \sum_{l=1}^{t}(R_l - \boldsymbol{A}_l^T\boldsymbol{\mu})^2 + \lambda\left\|\boldsymbol{\mu}\right\|_2^2,$$

where $\lambda \in \mathbb{R}_+$ control the scale of the regularization term. The estimator's analytical solution in time slot $t$ is

$$\hat{\boldsymbol{\mu}}_t = \boldsymbol{W}_t^{-1}\sum_{l=1}^{t}R_l\boldsymbol{A}_l,\quad \text{where } \boldsymbol{W}_t \triangleq \lambda\boldsymbol{I} + \sum_{l=1}^{t}\boldsymbol{A}_l\boldsymbol{A}_l^T.$$

To design the linear reward's UCB, we need the confidence set for estimated reward mean $\hat{\boldsymbol{\mu}}_{t-1}$ in the following lemma. This result is adapted from LinUCB literature [18].

*Lemma 2 ([18, Theorem 2]):* Denote the confidence set $\mathcal{C}_t$ as follows,

$$\mathcal{C}_t \triangleq \left\{\boldsymbol{\mu} \in \mathbb{R}^K : \|\hat{\boldsymbol{\mu}}_{t-1} - \boldsymbol{\mu}\|_{\boldsymbol{W}_{t-1}} \leq \sqrt{\beta_t}\right\}, \tag{10}$$

where $\|\boldsymbol{a}\|_{\boldsymbol{A}} \triangleq \boldsymbol{a}^T\boldsymbol{A}\boldsymbol{a}$ for $\boldsymbol{a} \in \mathbb{R}^K$ and $\boldsymbol{A} \in \mathbb{R}^{K\times K}$ is the norm induced by matrix $\boldsymbol{A}$ and

$$\sqrt{\beta_t} \triangleq \sqrt{\lambda}d + g\sqrt{2\log(1/\delta) + \log\left(\frac{\det(\boldsymbol{W}_{t-1})}{\lambda^K}\right)} \tag{11}$$

in which $g \in \mathbb{R}_+$ is the variance proxy of the sub-gaussian noise $\eta_t$. Then, the probability that there exists a time slot $t$ in $\mathbb{N}_+$ such that $\boldsymbol{\mu}^*$ lies inside $\mathcal{C}_t$ is no less than $1-\delta$, i.e.,

$$\mathbb{P}(\forall t \in \mathbb{N}^+, \boldsymbol{\mu}^* \in \mathcal{C}_t) \geq 1 - \delta.$$

The above lemma provides a *uniform* confidence bound for estimated $\hat{\boldsymbol{\mu}}_t$ for any time $t \in \mathbb{N}_+$. As the regret in (5) is for finite $T$ time slots, we set $\delta = 1/2\,T$ in later analysis. Noticing that the confidence set in Lemma 2 has an ellipsoidal form, we can derive the linear reward term's UCB as follows:

$$\max_{\boldsymbol{\mu}\in\mathcal{C}_t} R_t(\boldsymbol{A}) \leq \boldsymbol{A}^T\hat{\boldsymbol{\mu}}_{t-1} + \sqrt{\beta_t}\,\|\boldsymbol{A}\|_{\boldsymbol{W}_{t-1}^{-1}}. \tag{12}$$

We use the RHS of (12) as $\mathrm{UCB}_t^R(\boldsymbol{A})$'s formula (see the reasons at the end of **Deriving $\mathbf{LCB}_t^{\mathbf{fair}}(\boldsymbol{A})$**). Its detailed derivation is deferred to Section VIII-B.

## V. Regret Analysis

In the section, we derive a sublinear regret upper bound for the QLBF-UCB algorithm and a minimax regret lower bound for the QLBF model, both of which together reveal the optimality of the QLBF-UCB algorithm and the tightness of the regret upper bound.

### A. Regret Upper Bound

To assist the analysis of the regret upper bound of QLBF-UCB, we start from bounding the instantaneous regret in each single time slot, which is stated in the following lemma.

*Lemma 3:* With a probability of at least $1-\delta-1/2\,T$, the instantaneous regret $\mathrm{reg}_t \triangleq u_t(\boldsymbol{A}_t^*) - u_t(\boldsymbol{A}_t)$ in time slot $t$ satisfies

$$\mathrm{reg}_t \leq 3\,\|\boldsymbol{A}_t\|_{\boldsymbol{W}_{t-1}^{-1}}\sqrt{\beta_t} + 6\,s\left(\max_{k,t} w_{k,t}\right)^2\sqrt{\gamma_t},$$

where $\sqrt{\beta_t}$ and $\sqrt{\gamma_t}$ are parameters of reward confidence set $\mathcal{C}_t$ and arrival random variable's CDF's confidence band $\mathcal{F}_t$ respectively, and their definitions are in (11) and (7).

Lemma 3 states that the instantaneous regret $\mathrm{reg}_t$ is upper bounded by a linear combination of confidence set parameters $\sqrt{\beta_t}$ and $\sqrt{\gamma_t}$. As the time slot $t$ elapses, both the confidence set $\mathcal{C}_t$ and the confidence band $\mathcal{F}_t$ shrink, and thus the per time slot regret $\mathrm{reg}_t$ decreases. This implies that QLBF-UCB should converge to its benchmark policy. In the next theorem, we derive QLBF-UCB's cumulative regret upper bound.

*Theorem 1:* With a probability of $1-\delta-1/2\,T$, the regret of QLBF-UCB satisfies:

$$\begin{aligned}\mathrm{Reg}_T \leq\ & 6\sqrt{\max\{cd,1\}KT\log\left(\frac{K^2\lambda+Tc^2}{K^2\lambda}\right)} \\ & \times\left(\sqrt{\lambda}d+g\sqrt{2\log\frac{1}{\delta}+K\log\frac{K^2\lambda+Tc^2}{K^2\lambda}}\right) \\ & +12\,s\left(\max_{k,t} w_{k,t}\right)^2\sqrt{2T\log(2KT)},\end{aligned}$$

where $c$ is the limited resources of the server, $d$ is 2-norm bound of reward mean $\boldsymbol{\mu}^*$, i.e., $\|\boldsymbol{\mu}^*\|_2 \leq d$, $\lambda$ is the regularization factor in linear reward term's UCB, and $g$ is the variance proxy of reward's sub-gaussian noise.

Theorem 1 provides a sublinear regret upper bound for QLBF-UCB with probability $1-\delta-1/2\,T$. To derive an asymptotic expected regret upper bound for our algorithm, we select the $\delta=1/2\,T$ and take expectation of the regret over the probability $1-1/T$.

*Corollary 2:* Let $\delta=1/2\,T$, the regret of QLBF-UCB has the following asymptotic form

$$\mathbb{E}[\mathrm{Reg}_T] \leq O\left(\sqrt{T}\log T\right).$$

Corollary 2 states that the regret upper bound of QLBF-UCB has a sublinear order of $\sqrt{T}\log T$. If we consider the time average regret, i.e., $\frac{\mathbb{E}[\mathrm{Reg}_T]}{T}$, this regret upper bound becomes $O\left(\frac{\log T}{\sqrt{T}}\right)$. Notice that when $T$ goes to infinity, the average regret $\frac{\log T}{\sqrt{T}}$ would go to 0. This implies our online QLBF-UCB algorithm would converge to the offline benchmark policy.

### B. Regret Lower Bound

We provide a *minimax regret lower bound* defined as

$$\mathrm{Reg}_T^* \triangleq \inf_{\pi}\sup_{\boldsymbol{\mu},\mathcal{D}} \mathrm{Reg}_T^{\pi}(\boldsymbol{\mu},\mathcal{D})$$

for the QLBF model. This lower bound implies that for any given policy $\pi$, there exists an instance of the model (i.e., a pair of reward mean $\boldsymbol{\mu}^*$ and the arrival distribution $\mathcal{D}$), under which the regret of this policy is no less than the bound.

*Theorem 3:* Assume $K \leq 2\,T$, then there exists a reward mean vector $\boldsymbol{\mu}$ and an arrival distribution $\mathcal{D}$ such that

$$\mathrm{Reg}_T^* \geq \frac{c\sqrt{KT}}{16\sqrt{3}},$$

where $c$ is the server's finite capacity.

The main novelty in Theorem 3's proof is constructing a special environment (i.e., the reward mean $\boldsymbol{\mu}$ and arrival distribution $\mathcal{D}$) for our QLBF model which can signify the lower bound. This construction is different from the linear bandits' lower bound proof because one needs to take both the arrival distribution of the model and the nonlinear fairness term of the utility into consideration.

Compared with the $O(\sqrt{T}\log T)$ upper bound in Corollary 2, there is only a logarithmic factor absent in the lower bound in term of time horizon $T$. That implies both the near-optimality of our QLBF-UCB algorithm and the near-tightness of our upper bound analysis in Theorem 1.

## VI. Applications

In this section, we discuss several Internet-related applications of our QLBF model. Specifically, we provide a detailed discussion in three scenarios: 5G networks' bandwidth scheduling [7], mobile-edge computing nodes' resource scheduling [2], and crowdsourcing's task allocations [10]. We note that we focus on motivating the service reward of queues and the fairness among queues in the following application scenarios. Detailed modeling of each application requires additional effort, which is beyond the scope of this paper.

*Bandwidth Scheduling of 5G Wireless Networks:* In 2019, 5G (5th-generation) base stations started to provide commercial services to consumers [8]. 5G networks — with much higher capacities than 4 G network — are also required to efficiently support multiple kinds of traffic [7], e.g., Ultra Reliable Low Latency Communication (URLLC), enhanced Mobile Broadband (eMBB), etc. These traffic flows can form several transmission queues and these queues may have different priorities. For example, the URLLC traffic flow requires a very lower latency and any new arrivals in its corresponding queue should be served as soon as possible. These priorities can be mapped to the rewards of serving different queues in our QLBF model, and a 5G base station should aim to maximize its total transmission rewards. For another thing, although traffic from eMBB allows a little higher latency, the fairness among different traffic flows should also be take into consideration so that each flow can be transmitted timely. In this 5G networking application, the queues interarrival distribution $\mathcal{D}$ depends on different traffic flows, their service reward means $\boldsymbol{\mu}$ depend on the priorities of traffic flows, and the fairness is the variance of the lengths $\boldsymbol{L}_t$ of traffic flows.

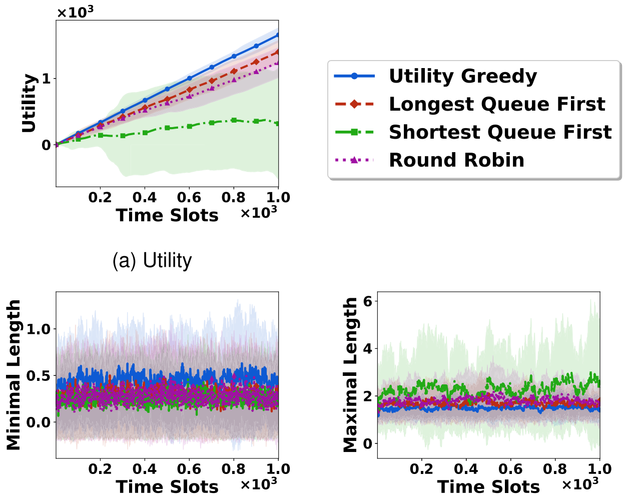
*Resources Scheduling of Mobile-Edge Computing Nodes:* Mobile-edge computing (MEC) is a key technique in nowadays networking systems, e.g., for Internet of Things (IoT) [2]. The key difference between MEC and cloud computing lies in the edge layer, which sits between the device layer, e.g., mobile phones, and the cloud layer, e.g., cloud servers. The edge computing nodes in this layer collect edge devices' data and computational tasks, execute these tasks with high speed, and send back the results to devices. This procedure can reduce edge devices' latency [4]. An application of QLBF in MEC is scheduling the edge computing node's service for mobile devices' offloading computational tasks [3]. Each edge devices' offloading tasks form a queue in its corresponding edge computing node. The edge node needs to decide how to serve each devices' offloading task queues with its limited computation capacity. Serving computational tasks of different devices may provides different rewards, e.g., a mobile phone's task is more urgent than a lamp's. For another thing, the edge node should also maintain the service fairness among these devices so that the latency of each devices can be reduced in some degree. In the MEC task scheduling, the queues interarrival distribution $\mathcal{D}$ depends on the IoT devices' features, their service reward means $\boldsymbol{\mu}$ are determined by the latency tolerance of the devices, and the fairness is computed as the variance of the offloading task lengths $\boldsymbol{L}_t$ of these devices.

*Task Allocation in Mobile Crowdsourcing:* Mobile Crowdsourcing is a powerful WWW approach [9], [10], [53] to obtaining services, collecting public opinions, etc. When a platform initiates a crowdsourcing task, a variety of individual workers may come. Workers from different social groups, e.g., age, gender, occupation, etc., constitute different queues and wait for the platform's task allocation. Workers in different groups may have different abilities in this specific task and thus their work outputs may have different qualities with respect to the platform. This task can be uniformly divided into multiple small tasks, each of which has a same and finite workload. Therefore, the platform should determine how to allocate its small tasks (e.g., in each time slot) to workers from different social groups so as to increase the task's final completion quality (e.g., total reward). For another thing, the platform should consider the fairness among different social groups. Because if some social groups are not assigned tasks, these types of workers may not come to the platform for future tasks, while for other kinds tasks in the future, these workers may be able to provide high quality outputs. Note that the number of workers and the finite amount of workloads can be regarded as dividable values as in QLBF when these quantities are large. In Crowdsourcing task allocation, the queues interarrival distribution $\mathcal{D}$ depends on the crowd characteristics of different social groups, their service reward means $\boldsymbol{\mu}$ are determined by the task output qualities of social groups, and the fairness is computed as the variance of the number of workers $\boldsymbol{L}_t$ of these social groups.

## VII. Evaluations

In this section, we conduct simulations to validate the effectiveness of the utility greedy policy and illustrate the QLBF-UCB's performance in both its sublinear regret guarantee and its performance over other scheduling policies.

We consider a QLBF model consisting of $K = 5$ queues (e.g., edge devices) and these queues' interarrivals are Bernoulli random variable with parameters $[0.5, 0.6, 0.7, 0.8, 0.9]$. By default, we set the total service capacity as the summation of arrival means plus $0.1$ times the summation of these queues' standard variances, i.e., $3.5 + 0.1\sigma \approx 3.715$. Note that the service capacity can be arbitrarily divided to serve jobs in queue. Each queue is associated with a service reward that is the same to its interarrival's mean. The fairness weight vector $\boldsymbol{w}_t$ is an all one vector $\mathbf{1}$, the fairness term's factor $s$ is set to 1, and the regularization scaler $\xi$ inside fairness is set to $0.1$. The scaler $\lambda$ of the regularized least square estimator for $\text{UCB}_t^R$ is set to 1. We run all simulations in 1000 time slots and the results are averaged over 50 rounds.



(a) Utility

(b) Minimal queue length

(c) Maximal queue length

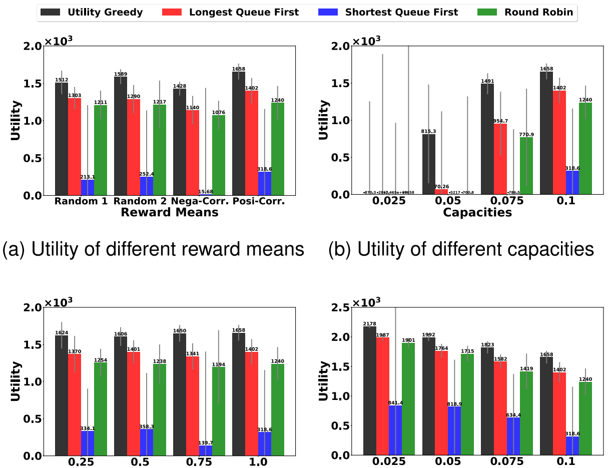
Fig. 1. Comparison of offline policies: in Fig. 1(a) the utility greedy policy outperforms other heuristic ones. Notably, in Fig. 1(b) and (c), the utility greedy policy has the highest minimal queue length and smallest maximal queue length. That implies the utility greedy policy enjoys a better fairness performance than other heuristic ones.

### A. The Effectiveness of Utility Greedy Policy

We validate the effectiveness of the benchmark policy in this subsection. With known model parameters, we compare the utility greedy policy with three scheduling policies: *shortest queue first*, *longest queue first* and *round-robin*. The shortest (longest) queue first policy sorts these queues according to their weighted queue lengths, i.e., $\boldsymbol{w}_t \circ \boldsymbol{L}_t$, into the ascending (descending) order and then assign the capacity according to the order. For the round-robin policy, we first arrange these queues in a circular order and then choose the start queue in turn at each time slot to allocate capacity. In all policies, we assign as much capacity as possible (i.e., up to its queue length) to each queue and the remaining capacity, if any, would be assigned to its next (according to queues' order) until there is no capacity or no requests left in queue.

Fig. 1 shows simulations under the default model parameters. Fig. 1(a) shows the utility greedy policy outperforms the other three heuristic policies. The utility greedy policy has the smallest queue length range because it has the largest minimal queue length in Fig. 1(b) and the smallest maximal queue length in Fig. 1(c). This implies that the greedy policy maintains a good fairness performance.

Then, we vary the default parameters to provide a more extensive comparison of these policies' performance in Fig. 2. Among all these experiments, the greedy policy (red bars) *always* outperforms the other policies. We start from altering these queues'



(a) Utility of different reward means (b) Utility of different capacities

(c) Utility of different $s$ factors (d) Utility of different $\xi$ factors

Fig. 2. Effectiveness of utility greedy policy: in various QLBF environments among all four sub-figures, the utility greedy policy always outperforms other heuristic policies. That implies the utility greedy policy is a good offline benchmark.

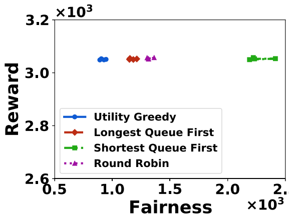
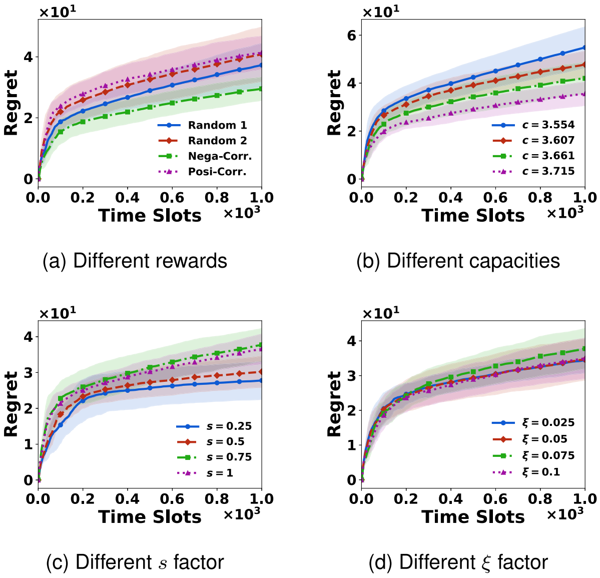


Fig. 3. Fairness *vs.* reward of offline policies when $s \in \{0.25, 0.5, 0.75, 1\}$ (as in Fig. 2(c)): offline policies' rewards are similar and for fairness, the utility greedy policy's is much better than others.

service rewards in Fig. 2(a). We call the default reward setting as "Posi-Cor." because the higher the queue's interarrival mean, the higher its service reward. We invert the reward means of these queue and name the case as "Nega-Corr." We also randomly select two service reward mean vector $[0.6, 0.8, 0.7, 0.9, 0.5]$ and $[0.7, 0.8, 0.9, 0.5, 0.6]$ to represent the general cases. In Fig. 2(b), we vary the default capacity $c = 3.5 + 0.1\sigma$'s interarrival variance factor 0.1 to $0.075, 0.05$ and $0.025$ to see the capacity's impact on the accumulative utilities of these policies. Next, we vary the fairness term's factor $s$ from the default 1.0 to $0.75, 0.5, 0.25$ to check how the different degrees of fairness consideration influences these policies in Fig. 2(c). We also separately plot the reward and fairness of Fig. 2(c)'s utilities in Fig. 3. This figure shows that the rewards of four policies are similar and the utility greedy policy enjoys a much better fairness than others. This implies that utility greedy policy is able to maximize the total reward while maintain the fairness. Lastly, in Fig. 2(c), we vary the fairness' regularization scaler $\xi$ from its default value 0.1 to $0.075, 0.05$ and $0.025$.

These extensive numerical simulations show that the greedy policy is a good benchmark policy as it is highly effective comparing with the other known heuristic policies.



(a) Different rewards (b) Different capacities

(c) Different $s$ factor (d) Different $\xi$ factor

Fig. 4. Regret of QLBF-UCB: QLBF-UCB enjoys a sublinear regret performance. That corroborates our regret upper bound in Theorem 1 and shows that the QLBF-UCB algorithm converges to the offline benchmark policy in a fast speed.

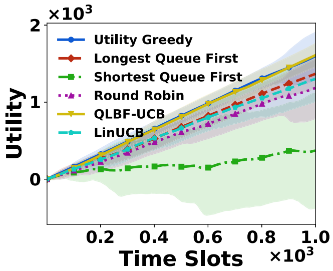


Fig. 5. QLBF-UCB vs. offline policies in utility: after time slot 600, the QLBF-UCB algorithms outperforms the longest queue first policy and the accumulative utility of QLBF-UCB is comparable to the utility greedy policy's which is the learning benchmark.

## *B. The Effectiveness of Our QLBF-UCB Algorithm*

In this subsection, we validate the QLBF-UCB algorithm's learning effectiveness on the utility greedy policy.

We start from validating QLBF-UCB's sublinear regret in learning the utility greedy policy in Fig. 4. Fig. 4's simulations are conducted under the same model parameters varying setting as that in Fig. 2. All these curves confirms the attractive sublinear regrets of QLBF-UCB.

We then compare the QLBF-UCB algorithm which needs to learn model parameters with pervious offline policies (which know model parameters). In Fig. 5, at the beginning (before time slot 200 around), the QLBF-UCB's performance (yellow) is not as good as the utility greedy policy (blue) and the longest queue first policy (red dashed). As time elapses, after time slot 600, the QLBF-UCB algorithms outperforms the longest queue first policy and its accumulative utility is comparable to the utility greedy policy's (blue) (which is our learning benchmark). We note that our QLBF-UCB algorithm also outperforms the

LinUCB (cyan) algorithm which only aims to maximize the total rewards.

From these simulations, we show that our proposed QLBF-UCB algorithm has a sublinear regret performance and can effectively learn the greedy benchmark policy effectively.

## VIII. Proofs of Lemmas and Theorems

### A. The Fairness Term's LCB Derivation

In this subsection, we show how to derive the nonlinear fairness term's LCB, i.e., $\text{LCB}_t^{\text{fair}}(\boldsymbol{A})$. We first present a proposition to show how to calculate the expected fairness term itself if the arrival distribution $\mathcal{D}$'s CDF is given. Next, we replace the fairness calculation formula's CDF with CDF's upper and lower confidence functions constructed by Lemma 1 accordingly and obtain an explicit calculation formula for $\text{LCB}_t^{\text{fair}}(\boldsymbol{A})$.

The following proposition states a formula to calculate the fairness term $\mathbb{E}_{\boldsymbol{\Lambda}_t \sim F} \bar{V}_t(\boldsymbol{\Lambda}_t)$ from the CDF of arrival distribution $\mathcal{D}$.

*Proposition 1:* Given the CDF function $F$ of the arrival distribution $\mathcal{D}$, the expected fairness in time slot $t$ can be calculated as follows

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\Lambda}_t \sim F} \bar{V}_t(\boldsymbol{\Lambda}_t) \\
&= \bar{V}_t(b \cdot \mathbf{1}) - \sum_{i=1}^{K} \int_0^b F^{(k)}(\Lambda_{i,t}) \frac{\partial \bar{V}_t(b, \ldots, b, \Lambda_{i,t}, b, \ldots, b)}{\partial \Lambda_{i,t}} d\Lambda_{i,t} \\
&\quad + \sum_{1 \le i < j \le K} \int_0^b \int_0^b F^{(k)}(\Lambda_{i,t}) F^{(j)}(\Lambda_{j,t}) \\
&\quad \times \frac{\partial^2 \bar{V}_t(b, \ldots, b, \Lambda_{i,t}, b, \ldots, b, \Lambda_{j,t}, b, \ldots, b)}{\partial \Lambda_{j,t} \partial \Lambda_{i,t}} d\Lambda_{i,t} d\Lambda_{j,t}.
\end{aligned}
$$

*Proof of Proposition 1:* We prove the proposition via induction with respect to the number of queues $K$.

When $K$ equals to 1, i.e., $\boldsymbol{\Lambda}_t \in \mathbb{R}$ is a one dimensional random variable (supported in $[0, b]$), we have

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\Lambda}_t \sim F} \bar{V}_t(\boldsymbol{\Lambda}_t) &= \int_0^b \bar{V}_t(\Lambda_{1,t}) dF^{(1)}(\Lambda_{1,t}) \\
&= \bar{V}_t(\Lambda_{1,t}) F^{(1)}(L_t)|_{\Lambda_{1,t}=0}^{b} \\
&\quad - \int_0^b F^{(1)}(\Lambda_{1,t}) d\bar{V}_t(\Lambda_{1,t}) \\
&= \bar{V}_t(b) - \int_0^b F^{(1)}(\Lambda_{1,t}) \frac{\partial \bar{V}_t(\Lambda_{1,t})}{\partial \Lambda_{1,t}} d\Lambda_{1,t},
\end{aligned}
$$

which fits the proposition.

Suppose when $K = m-1$, the proposition holds. Then we show that the proposition still holds when $K = m$.

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\Lambda}_t \sim F} \bar{V}_t(\boldsymbol{\Lambda}_t) \\
&= \underbrace{\int_0^b \int_0^b \cdots \int_0^b}_{m-1} \bar{V}_t(\boldsymbol{\Lambda}_t) \underbrace{dF^{(1)}(\Lambda_{1,t}) \ldots dF^{(m-1)}(\Lambda_{m-1,t})}_{m-1} \\
&\quad \times dF^{(m)}(\Lambda_{m,t}) \\
&= \int_0^b \Bigg[ \bar{V}_t(b, \ldots, b, \Lambda_{m,t}) \\
&\quad - \sum_{i=1}^{m-1} \int_0^b F^{(i)}(\Lambda_{i,t}) \frac{\partial \bar{V}_t(b, \ldots, b, \Lambda_{i,t}, b, \ldots, b, \Lambda_{m,t})}{\partial \Lambda_{i,t}} d\Lambda_{i,t} \\
&\quad + \sum_{1 \le i < j \le m-1} \int_0^b \int_0^b F^{(i)}(\Lambda_{i,t}) F^{(j)}(\Lambda_{j,t}) \\
&\quad \times \frac{\partial^2 \bar{V}_t(b, \ldots, b, \Lambda_{i,t}, b, \ldots, b, \Lambda_{j,t}, b, \ldots, b, \Lambda_{m,t})}{\partial \Lambda_{j,t} \partial \Lambda_{i,t}} \\
&\quad \times d\Lambda_{i,t} d\Lambda_{j,t} \Bigg] dF^{(m)}(\Lambda_{m,t}) \\
&= \bar{V}_t(b \cdot \mathbf{1}) - \sum_{i=1}^{m-1} \int_0^b F^{(i)}(\Lambda_{i,t}) \frac{\partial \bar{V}_t(b, \ldots, b, \Lambda_{i,t}, b, \ldots, b)}{\partial \Lambda_{i,t}} d\Lambda_{i,t} \\
&\quad + \sum_{1 \le i < j \le m-1} \int_0^b \int_0^b F^{(i)}(\Lambda_{i,t}) F^{(j)}(\Lambda_{j,t}) \\
&\quad \times \frac{\partial^2 \bar{V}_t(b, \ldots, b, \Lambda_{i,t}, b, \ldots, b, \Lambda_{j,t}, b, \ldots, b)}{\partial \Lambda_{j,t} \partial \Lambda_{i,t}} d\Lambda_{i,t} d\Lambda_{j,t} \\
&\quad + \int_0^b F^{(m)}(\Lambda_{m,t}) \Bigg[ \frac{\partial \bar{V}_t(b, \ldots, b, b)}{\partial \Lambda_{m,t}} - \sum_{i=1}^{m-1} \int_0^b F^{(i)}(\Lambda_{i,t}) \\
&\quad \times \frac{\partial^2 \bar{V}_t(b, \ldots, \Lambda_{i,t}, \ldots, b, \Lambda_{m,t})}{\partial \Lambda_{m,t} \partial \Lambda_{i,t}} d\Lambda_{i,t} \Bigg] d\Lambda_{m,t} \\
&= \bar{V}_t(b \cdot \mathbf{1}) - \sum_{i=1}^{m} \int_0^b F^{(i)}(\Lambda_{i,t}) \frac{\partial \bar{V}_t(b, \ldots, b, \Lambda_{i,t}, b, \ldots, b)}{\partial \Lambda_{i,t}} d\Lambda_{i,t} \\
&\quad + \sum_{1 \le i < j \le m} \int_0^b \int_0^b F^{(i)}(\Lambda_{i,t}) F^{(j)}(\Lambda_{j,t}) \\
&\quad \times \frac{\partial^2 \bar{V}_t(b, \ldots, b, \Lambda_{i,t}, b, \ldots, b, \Lambda_{j,t}, b \ldots, b)}{\partial \Lambda_{j,t} \partial \Lambda_{i,t}} d\Lambda_{i,t} d\Lambda_{j,t},
\end{aligned}
$$

where we utilize the supposition in $K = m-1$ case in the second equation, the third equation applies integral by part and the property that variance function's third and above derivatives are zero. This validates that the proposition holds when $K = m$. By the principle of induction, we conclude the proof.

To derivate the $\text{LCB}_t^{\text{fair}}(\boldsymbol{A})$'s lower bound in (9), we calculate the fairness term $\mathbb{E}_{\boldsymbol{\Lambda}_t \sim \hat{F}_{t-1}} \bar{V}_t(\boldsymbol{\Lambda}_t)$ by Proposition 1 and separately bound its two integral terms as follows.

To bound $\int_0^b \hat{F}^{(k)}(\Lambda_{i,t}) \frac{\partial \bar{V}_t(b, \ldots, b, \Lambda_{i,t}, b, \ldots, b)}{\partial \Lambda_{i,t}} d\Lambda_{i,t}$, we note that the partial derivative inside the integral is equal to

$$
\begin{aligned}
&2w_{i,t}^2(L_{i,t-1} + \Lambda_{i,t} - A_{i,t}) + \frac{2w_{i,t}}{K}(w_{i,t}(L_{i,t-1} + \Lambda_{i,t} - A_{i,t}) \\
&\qquad - \frac{1}{K} \sum_{j=1}^{K} (w_{j,t}(L_{j,t-1} + \Lambda_{i,t} - A_{i,t}))).
\end{aligned}
$$

It is an increasing linear function and we denote the derivative's unique *zero* as $z_{k,t}$. Define $q_{k,t} \triangleq b \mathbb{1}_{\{z_{k,t} \ge b\}} + z_{k,t} \mathbb{1}_{\{0 < z_{k,t} < b\}}$. We can separate the integral $\int_0^b$ to two parts $\int_0^{q_{k,t}}$ and $\int_{q_{k,t}}^b$.

When $\Lambda_{i,t} \le q_{k,t}$ (resp. $\Lambda_{i,t} > q_{k,t}$), the derivative is negative (resp. positive) and we can replace the $\hat{F}_{t-1}^{(i)}$ by its lower bound $L_{t-1}^{(k)}$ (resp. upper bounded by $U_{t-1}^{(k)}$).

To bound the double integral term at last, we note that the partial second derivative of $\bar{V}_t$ with respect to two different arrivals is

$$\frac{\partial^2 \bar{V}_t(b,\ldots,b,\Lambda_{i,t},b,\ldots,b,\Lambda_{j,t},b,\ldots,b)}{\partial\Lambda_{j,t}\partial\Lambda_{i,t}} = -\frac{2w_{i,t}w_{j,t}}{K^2},$$

which is always negative. So, we can replace $\hat{F}^{(k)}(\Lambda_{i,t})\hat{F}^{(j)}(\Lambda_{j,t})$ by its upper bound $U^{(k)}(\Lambda_{i,t})U^{(j)}(\Lambda_{j,t})$.

### B. Reward Term's UCB Derivation

Recall that the confidence set of reward mean $\boldsymbol{\mu}^*$ in (10) has an ellipsoidal form and can be rewritten as follows:

$$\begin{aligned}\mathcal{C}_t &= \left\{\boldsymbol{\mu} \in \mathbb{R}^K : \|\hat{\boldsymbol{\mu}}_{t-1} - \boldsymbol{\mu}\|_{\boldsymbol{W}_{t-1}} \le \sqrt{\beta_t}\right\} \\ &= \left\{\hat{\boldsymbol{\mu}}_{t-1} + \sqrt{\beta_t}\boldsymbol{W}_{t-1}^{-\frac{1}{2}}\boldsymbol{x} : \|\boldsymbol{x}\|_2 \le 1, \boldsymbol{x} \in \mathbb{R}^K\right\},\end{aligned}$$

where the matrix $\boldsymbol{W}_{t-1}^{-\frac{1}{2}}$ is the "square root" of the inverse matrix $\boldsymbol{W}_{t-1}^{-1}$, i.e., $\boldsymbol{W}_{t-1}^{-1} = \boldsymbol{W}_{t-1}^{-\frac{1}{2}} \cdot \boldsymbol{W}_{t-1}^{-\frac{1}{2}}$.

So, the $\text{UCB}_t^R(\boldsymbol{A})$ defined as $\max_{\boldsymbol{\mu}\in\mathcal{C}_t} \boldsymbol{A}^T\boldsymbol{\mu}$ can be calculated as follows:

$$\begin{aligned}\text{UCB}_t^R(\boldsymbol{A}) &= \max_{\boldsymbol{\mu}\in\mathcal{C}_t} \boldsymbol{A}^T\boldsymbol{\mu} \\ &= \boldsymbol{A}^T\hat{\boldsymbol{\mu}}_{t-1} + \max_{\|\boldsymbol{x}\|_2\le 1} \sqrt{\beta_t}\boldsymbol{A}^T\boldsymbol{W}_{t-1}^{-\frac{1}{2}}\boldsymbol{x} \\ &= \boldsymbol{A}^T\hat{\boldsymbol{\mu}}_{t-1} + \sqrt{\beta_t}\,\|\boldsymbol{A}\|_{\boldsymbol{W}_{t-1}^{-1}},\end{aligned}$$

where the last equality holds when choosing $\boldsymbol{x} = \frac{\boldsymbol{W}_{t-1}^{-\frac{1}{2}}\boldsymbol{A}}{\|A\|_{\boldsymbol{W}_{t-1}^{-1}}}$.

### C. Proof of Lemma 1

We first state the *Dvoretzky-Kiefer-Wolfowitz* inequality for one dimension random variable in the following lemma. Based on the random variable's empirical cumulative distribution function $\hat{F}_t(x)$, this lemma constructs a confidence band that contains the true CDF function $F(x)$ with the confidence $1-\delta$.

*Lemma 4 ([19, Theorme 7.1]):* Denote $\epsilon_t = \sqrt{\frac{1}{2t}\log\frac{2}{\alpha}}$. The true CDF function $F(x)$ is inside the confidence band

$$\begin{aligned}&\{f \in C_b : \\ &\max\{\hat{F}_t(x) - \epsilon_t, 0\} \le f(x) \le \min\{\hat{F}_t(x) + \epsilon_t, 1\}, \forall x\}\end{aligned}$$

with a confidence of $1-\alpha$. That is,

$$\begin{aligned}\mathbb{P}(\max\{\hat{F}_t(x) - \epsilon_t, 0\} \le F(x) \le \min\{\hat{F}_t(x) + \epsilon_t, 1\}, \forall x) \\ \ge 1-\alpha.\end{aligned}$$

Denote the event $E_t \triangleq \{\max\{\hat{F}_t(x) - \epsilon_t, 0\} \le F(x) \le \min\{\hat{F}_t(x) + \epsilon_t, 1\}, \forall x\}$. Substituting $\alpha$ with $\alpha_t \triangleq \frac{1}{K(t+T)^2}$ and the band distance $\epsilon_t$ with $\sqrt{\gamma_t}$ in Lemma 4, we have $\mathbb{P}(\neg E_t) \le \alpha_t$. Then, we apply union bound to derive a confidence band holding for all $t \in \{1,\ldots,T\}$,

$$\begin{aligned}\mathbb{P}(\exists t \le T, \neg E_t) &\le \sum_{t=1}^{T}\mathbb{P}(\neg E_t) \le \sum_{t=1}^{n}\alpha_t \\ &\le \int_{t=1}^{T}\frac{1}{K(t+T)^2}dt \le \frac{1}{2KT}.\end{aligned}$$

### D. Proof of Lemma 3

We start from bounding the per time regret as follows

$$\begin{aligned}\text{reg}_t &= u_t(\boldsymbol{A}_t^*) - u_t(\boldsymbol{A}_t)) \\ &\le \text{UCB}(\boldsymbol{A}_t^*) - u_t(\boldsymbol{A}_t) \le \text{UCB}(\boldsymbol{A}_t) - u_t(\boldsymbol{A}),\end{aligned} \tag{13}$$

where the first inequality is from UCB's definition in (6) which holds with probability $(1-\delta)(1-1/2\,T)$, and the second is from the utility greedy action's definition.

Then, we substitute the reward's UCB expression in (12) and fairness' LCB expression in (9) into (13)'s RHS and get

$$\begin{aligned}&\text{UCB}(\boldsymbol{A}_t) - u_t(\boldsymbol{A}_t) \\ &= \text{UCB}_t^R(\boldsymbol{A}_t) - \boldsymbol{A}_t^T\boldsymbol{\mu}^* - s(\text{LCB}_t^{\text{fair}}(\boldsymbol{A}_t) - \mathbb{E}_{\boldsymbol{\Lambda}_t\sim\hat{F}_{t-1}}\bar{V}_t(\boldsymbol{\Lambda}_t)) \\ &= \boldsymbol{A}_t^T\hat{\boldsymbol{\mu}}_t + \sqrt{\boldsymbol{\beta}}_t\,\|\boldsymbol{A}_t\|_{\boldsymbol{W}_{t-1}^{-1}} - \boldsymbol{A}_t^T\boldsymbol{\mu}^* \\ &\quad - s\sum_{i=1}^{K}\Bigg[\int_0^{q_{i,t}} L_{t-1}^{(i)}(\Lambda_{i,t})\frac{d\bar{V}_t(b,\ldots,b,\Lambda_{i,t},b,\ldots,b)}{d\Lambda_{i,t}}d\Lambda_{i,t} \\ &\quad + \int_{q_{i,t}}^{b} U_{t-1}^{(i)}(\Lambda_{i,t})\frac{d\bar{V}_t(b,\ldots,b,\Lambda_{i,t},b,\ldots,b)}{d\Lambda_{i,t}}d\Lambda_{i,t} \\ &\quad - \int_0^{b} F_{t-1}^{(i)}(\Lambda_{i,t})\frac{d\bar{V}_t(b,\ldots,b,\Lambda_{i,t},b,\ldots,b)}{d\Lambda_{i,t}}d\Lambda_{i,t}\Bigg] \\ &\quad + s\sum_{1\le i<j\le K}\left(-\frac{2w_{i,t}w_{j,t}}{K^2}\right) \\ &\quad \times \Bigg[\int_0^b U_{t-1}^{(i)}(\Lambda_{i,t})d\Lambda_{i,t}\int_0^b U_{t-1}^{(i)}(\Lambda_{j,t})d\Lambda_{j,t} \\ &\quad - \int_0^b F_{t-1}^{(i)}(\Lambda_{i,t})d\Lambda_{i,t}\int_0^b F_{t-1}^{(i)}(\Lambda_{j,t})d\Lambda_{j,t}\Bigg] \\ &\le 3\sqrt{\beta_t}\,\|\boldsymbol{A}_t\|_{\boldsymbol{W}_{t-1}^{-1}} + s\sum_{1\le i<j\le K}\left(-\frac{2w_{i,t}w_{j,t}}{K^2}\right) \\ &\quad \times \left[\sqrt{\gamma_t}\int_0^b [F_{t-1}^{(i)}(x) + F_{t-1}^{(j)}(x)]dx + \gamma_t\right] \\ &\le 3\sqrt{\beta_t}\,\|\boldsymbol{A}_t\|_{\boldsymbol{W}_{t-1}^{-1}} + 6s\left[\max_k w_{k,t}\right]^2\sqrt{\gamma_t},\end{aligned}$$

where the $q_{i,t}$ is defined at the end of Section VIII-A, and in the first inequality we omit the forth term (the $\sum_{i=1}^K$ summation term) because it is subtracting a positive term.

### *E. Proof of Theorem 1*

Lemma 3 divides the instantaneous per time regret into two terms: $3\|\boldsymbol{A}_t\|_{\boldsymbol{W}_{t-1}^{-1}}\sqrt{\beta_t}$ and $6[\max_i w_{i,t}]^2\sqrt{\gamma_t}$. We separately bound both.

We adapt the standard LinUCB's regret result to the first term (Theorem 19.2 in [54]), and get the following

$$\sum_{t=1}^{T} 3\left\|\boldsymbol{A}_t\right\|_{\boldsymbol{W}_{t-1}^{-1}}\sqrt{\beta_t}\le 6\sqrt{\max\{cd,1\}Kn\log\left(\frac{K^2\lambda+Tc^2}{K^2\lambda}\right)}$$
$$\times\left(\sqrt{\lambda}d+g\sqrt{2\log\frac{1}{\delta}+K\log\frac{K^2\lambda+Tc^2}{K^2\lambda}}\right).$$

Note that the summation of $\sqrt{\gamma_t}$ can be bounded as follows.

$$\begin{aligned}\sum_{t=1}^{T}\sqrt{\gamma_t}&=\sum_{t=1}^{T}\sqrt{\frac{1}{t}\log(2K(t+T)^2)}\\&\le\sum_{t=1}^{T}\sqrt{\frac{2}{t}\log[K(t+T)]}\le\sum_{t=1}^{T}\sqrt{\frac{2}{t}\log(2KT)}\\&=\sqrt{2\log(2KT)}\sum_{t=1}^{T}\frac{1}{\sqrt{t}}\le 2\sqrt{2T\log(2KT)}.\end{aligned}$$

We then bound the second part $\sum_{t=1}^{T}6[\max_i w_{k,t}]^2\sqrt{\gamma_t}\le 12[\max_{k,t}w_{k,t}]^2\sqrt{2T\log(2KT)}$. The above two summation bounds together lead to the theorem.

### *F. Proof of Theorem 3*

To illustrate the minimax lower bound, we construct a potential worst case environment in QLB-F model.

Let $\Delta\triangleq\frac{\sqrt{K/T}}{4\sqrt{3}}$ and $\boldsymbol{\mu}\in\{\pm\Delta\}^K$, and for $i\in\mathcal{K}$ define

$$\tau_i\triangleq T\wedge\min\left\{t:\sum_{s=1}^{t}A_{i,s}^2\ge\frac{T}{K}\right\}$$

where $a\wedge b=\min\{a,b\}$. The reward mean vector $\boldsymbol{\mu}$ can be chosen in $\{\pm\Delta\}^K$ and without loss of generality we assume that there is at least one $k\in\mathcal{K}$ such that $\mu_k=\Delta$. For the fairness part, let the initial queue length be $\boldsymbol{L}_0=l\cdot\boldsymbol{1}$, where $l\in\mathbb{R}_+$ is large enough such that the queue length constraint for action is vacuum (i.e., the available action set has only the capacity constraint $\|\boldsymbol{A}\|_1\le c$). There also exists a corresponding arrival distribution (in each time slot, there are $c$ constant new arrivals in arm $k$ and zero arrival in other arms) such that the fairness term is always a constant, i.e., $\boldsymbol{L}_t=l\cdot\boldsymbol{1}$ always.

In the above environment, the optimal action is to assign all capacity $c$ to the arm $k$ in all time slots. As the distribution $\mathcal{D}$ adheres to reward mean vector, our parameter space depends only on $\boldsymbol{\mu}$. The following proof mechanism is similar to LinUCB's lower bound in an unit ball action space in Theorem 24.2 [54]. With the special construction, we have

$$\begin{aligned}\mathrm{Reg}_T^*(\boldsymbol{\mu},\mathcal{D})&\ge\mathbb{E}_{\boldsymbol{\mu},\mathcal{D}}\left[\sum_{t=1}^{T}\left(c\Delta-\sum_{i=1}^{K}A_{i,t}\mu_i\right)\right]\\&=\frac{c\Delta}{\sqrt{K}}\mathbb{E}_{\boldsymbol{\mu},\mathcal{D}}\left[\sum_{t=1}^{T}\sum_{i=1}^{K}\left(\frac{1}{\sqrt{K}}-\frac{A_{i,t}}{c\sqrt{K}}\mathrm{sign}(\mu_i)\right)\right]\\&\ge\frac{c\Delta}{2}\mathbb{E}_{\boldsymbol{\mu},\mathcal{D}}\left[\sum_{t=1}^{T}\sum_{i=1}^{K}\left(\frac{1}{\sqrt{K}}-\frac{A_{i,t}}{c\sqrt{K}}\mathrm{sign}(\mu_i)\right)^2\right]\\&\ge\frac{c\Delta}{2}\sum_{i=1}^{K}\mathbb{E}_{\boldsymbol{\mu},\mathcal{D}}\\&\quad\times\left[\sum_{t=1}^{\tau_i}\left(\frac{1}{\sqrt{K}}-\frac{A_{i,t}}{c\sqrt{K}}\mathrm{sign}(\mu_i)\right)^2\right],\end{aligned}$$

where the first inequality holds as the optimal action's fairness penalty is zero, and the second is by $\|\boldsymbol{A}_t/c\sqrt{K}\|_2^2\le 1$.

For any $i\in\mathcal{K}$ and $x\in\{\pm 1\}$, define $H_i(x)=\sum_{t=1}^{\tau_i}(1/\sqrt{K}-A_{i,t}x)^2$ and select $\boldsymbol{\mu}'\in\{\pm\Delta\}^K$ such that $\mu'_j=\mu_j$ for any $j\neq i$ and $\mu'_i=-\mu_i$. Then, we adapt Theorem 24.2 [54]'s intermediate result to our case as follows

$$\mathbb{E}_{\boldsymbol{\mu},\mathbb{D}}[H_i(1)]+\mathbb{E}_{\boldsymbol{\mu},\mathcal{D}}[H_i(-1)]\ge\frac{T}{K}.$$

We apply the average hammer over any possible arm $i\in\mathcal{K}$ whose $\mu_i=\Delta$,

$$\begin{aligned}&\sum_{\boldsymbol{\mu}\in\{\pm\Delta\}^K,\mathcal{D}}\mathrm{Reg}_T^*(\boldsymbol{\mu},\mathcal{D})\\&\ge\frac{\Delta\sqrt{K}}{2}\sum_{i=1}^{K}\sum_{\boldsymbol{\mu}\in\{\pm\Delta\}^K}\mathbb{E}_{\boldsymbol{\mu}}[H_i(\mathrm{sign}(\mu_i))]\\&=\frac{\Delta\sqrt{K}}{2}\sum_{i=1}^{K}\sum_{\boldsymbol{\mu}_{-i}\in\{\pm\Delta\}^{K-1}}\sum_{\mu_i\in\{\pm\Delta\}}\mathbb{E}_{\boldsymbol{\mu}}[H_i(\mathrm{sign}(\mu_i))]\\&\ge\frac{\Delta\sqrt{K}}{2}\sum_{i=1}^{K}\sum_{\boldsymbol{\mu}_{-i}\in\{\pm\Delta\}^{K-1}}\frac{T}{K}=2^{K-2}c\Delta T.\end{aligned}$$

Finally, we substitute $\Delta$ back to $\frac{\sqrt{K/T}}{4\sqrt{3}}$. That shows there exists a pair of $\boldsymbol{\mu}$ and $\mathcal{D}$ such that $\mathrm{Reg}_T^*(\boldsymbol{\mu},\mathcal{D})\ge c\sqrt{KT}/(16\sqrt{3})$.

## IX. Conclusion

In this paper, we proposed QLBF, an online learning model derived from many queueing applications, e.g., mobile computing nodes' resources and network bandwidth scheduling, manufacturing assignment, etc. The QLBF model abstracts a system consisting of a resources allocator (server) with dividable and limited capacities, and multiple queues whose unit service rewards and arrival distributions are all unknown. To maximize the reward and maintain the fairness from serving these queues, we construct a utility function containing a linear reward term and a nonlinear fairness term. The model's fairness is measured by the variance of queue lengths in the system, which can represent the fairness in many applications. We choose a utility greedy offline policy as the learning benchmark, and focus on the QLBF model's online learnability. To maximize the total utility in an online environment, we designed the QLBF-UCB algorithm that is based on deriving upper confidence bounds (UCBs) for both the reward and the fairness terms. The algorithm enjoys a sublinear regret upper bound that is close to the model's regret lower bound, which reveals that the designed algorithm is nearly optimal and both upper and lower bounds are nearly tight. We also discuss several potential applications of QLBF model including mobile edge computing, 5G wireless network, and mobile crowdsourcing. Finally, we conduct simulations to validate the greedy policy's effectiveness in offline environment and our online QLBF-UCB algorithm's performance in learning the greedy policy.

The utility function contains a factor $s$ balancing the importance of reward and fairness. Although, practically, one can start from several choices of $s$ and then select the one with good performance, how to select a proper $s$ factor to avoid overbalancing with theoretical guarantee is a interesting future direction. Besides maximizing the integrated utility function alone, another approach to tackling the linear reward and nonlinear fairness terms is the multi-objective optimization. For example, one can study the Pareto optimality [55] of both objectives in this problem and devise algorithms to achieve this optimality. We leave this direction as a potential future work. Another interesting future work is to consider the case of the time-dependent reward mean $\boldsymbol{\mu}_t$ which covers more realistic applications. One can model it as the non-stationary bandits with change points problem and extend our current QLBF-UCB algorithm to address it. The high-level idea of the extension of LinUCB to non-stationary environment [56] might be helpful for this extension.

## References

[1] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory*, 2016, pp. 1451–1455.

[2] S. Wan, J. Lu, P. Fan, and K. B. Letaief, "Intelligent networking with mobile edge computing: Vision and challenges for dynamic network scheduling," 2020, *arXiv: 2004.13926*.

[3] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[4] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[5] C.-P. Li, J. Jiang, W. Chen, T. Ji, and J. Smee, "5G ultra-reliable and low-latency systems design," in *Proc. IEEE Eur. Conf. Netw. Commun.*, 2017, pp. 1–5.

[6] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 692–704, Feb. 2019.

[7] A. Anand, G. De Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 477–490, Apr. 2020.

[8] A. Narayanan et al., "A first look at commercial 5G performance on smartphones," in *Proc. Web Conf.*, 2020, pp. 894–905.

[9] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," *Commun. ACM*, vol. 54, no. 4, pp. 86–96, 2011.

[10] D. Deng, C. Shahabi, and L. Zhu, "Task matching and scheduling for multiple workers in spatial crowdsourcing," in *Proc. Int. Conf. Adv. Geographic Inf. Syst.*, 2015, pp. 1–10.

[11] J. Kim and W. Lee, "Stochastic decision making for adaptive crowdsourcing in medical big-data platforms," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 45, no. 11, pp. 1471–1476, Nov. 2015.

[12] S. R. Chakravarthy and A. N. Dudin, "A queueing model for crowdsourcing," *J. Oper. Res. Soc.*, vol. 68, no. 3, pp. 221–236, 2017.

[13] D. Bertsekas and J. N. Tsitsiklis, "Introduction to probability," *Athena Sci.*, vol. 1, 2008.

[14] K. Brunnström et al., "Qualinet white paper on definitions of quality of experience," 2007, *arXiv:2007.07032*.

[15] S. M. Ross, *Stochastic Processes*. Hoboken, NJ, USA: Wiley, 1996, vol. 2.

[16] J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris, *Fundamentals of Queueing Theory*. Hoboken, NJ, USA: Wiley, 2018.

[17] P. Auer, "Using confidence bounds for exploitation-exploration tradeoffs," *J. Mach. Learn. Res.*, vol. 3, pp. 397–422, 2002.

[18] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2312–2320.

[19] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference*. Berlin, Germany: Springer, 2004.

[20] P. Jacko, "Restless bandits approach to the job scheduling problem and its extensions," *Modern Trends Controlled Stochastic Processes: Theory Appl.*, pp. 248–267, 2010.

[21] S. Krishnasamy, R. Sen, R. Johari, and S. Shakkottai, "Regret of queueing bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1669–1677.

[22] T. Stahlbuhk, B. Shrader, and E. Modiano, "Learning algorithms for minimizing queue length regret," in *Proc. IEEE Int. Symp. Inf. Theory*, 2018, pp. 1001–1005.

[23] M. Joseph, M. Kearns, J. H. Morgenstern, and A. Roth, "Fairness in learning: Classic and contextual bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 325–333.

[24] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1799–1813, Second Quarter 2020.

[25] X. Zhang and M. Liu, "Fairness in learning-based sequential decision algorithms: A survey," 2020, *arXiv: 2001.04861*.

[26] I. Bistritz, T. Z. Baharav, A. Leshem, and N. Bamobs, "My fair bandit: Distributed learning of max-min fairness with multi-player bandits," 2020, *arXiv: 2002.09808*.

[27] V. Patil, G. Ghalme, V. Nair, and Y. Narahari, "Achieving fairness in the stochastic multi-armed bandit problem," *J. Mach. Learn. Res.*, vol. 22, pp. 174–1, 2021.

[28] Y. Chen, A. Cuellar, H. Luo, J. Modi, H. Nemlekar, and S. Nikolaidis, "Fair contextual multi-armed bandits: Theory and experiments," in *Proc. Conf. Uncertainty Artif. Intell.*, 2020, pp. 181–190.

[29] W. Xia, T. Q. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7108–7123, Nov. 2020.

[30] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1552–1564, Jul. 2021.

[31] R. K. Jain et al., "A quantitative measure of fairness and discrimination," Eastern Research Lab., Digital Equipment Corporation, Hudson, MA, USA, vol. 21, 1984.

[32] T. Hoßfeld, L. Skorin-Kapov, P. E. Heegaard, and M. Varela, "Definition of QoE fairness in shared systems," *IEEE Commun. Lett.*, vol. 21, no. 1, pp. 184–187, Jan. 2017.

[33] A. Sani, A. Lazaric, and R. Munos, "Risk-aversion in multi-armed bandits," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 3275–3283.

[34] S. Vakili and Q. Zhao, "Risk-averse multi-armed bandit problems under mean-variance measure," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 6, pp. 1093–1111, Sep. 2016.

[35] L. Wang, Y. Bai, W. Sun, and T. Joachims, "Fairness of exposure in stochastic bandits," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10 686–10 696.

[36] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, no. 1, pp. 4–22, 1985.

[37] J. Gittins, K. Glazebrook, and R. Weber, *Multi-Armed Bandit Allocation Indices*. Hoboken, NJ, USA: Wiley, 2011.

[38] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2/3, pp. 235–256, 2002.

[39] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic linear optimization under bandit feedback," in *Proc. Annu. Conf. Learn. Theory*, 2008, pp. 355–366.

[40] P. Rusmevichientong and J. N. Tsitsiklis, "Linearly parameterized bandits," *Math. Operations Res.*, vol. 35, no. 2, pp. 395–411, 2010.

[41] J. Langford and T. Zhang, "The epoch-greedy algorithm for multi-armed bandits with side information," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 817–824.

[42] S. Agrawal and N. Devanur, "Linear contextual bandits with knapsacks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3450–3458.

[43] A. Kazerouni, M. Ghavamzadeh, Y. A. Yadkori, and B. Van Roy, "Conservative contextual linear bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3910–3919.

[44] B. Hao, T. Lattimore, and C. Szepesvari, "Adaptive exploration in linear contextual bandit," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 3536–3545.

[45] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities: A Nonasymptotic Theory Independence*, Oxford, U.K.: Oxford Univ. Press, 2013.

[46] J. Wong, J. Sauve, and J. Field, "A study of fairness in packet-switching networks," *IEEE Trans. Commun.*, vol. 30, no. 2, pp. 346–353, Feb. 1982.

[47] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using openflow-assisted adaptive video streaming," in *Proc. Workshop Future Human-Centric Multimedia Netw.*, 2013, pp. 15–20.

[48] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, 2000.

[49] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, 2005, pp. 385–394.

[50] E. Hazan and K. Levy, "Bandit convex optimization: Towards tight bounds," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 784–792.

[51] A. S. Suggala, P. Ravikumar, and P. Netrapalli, "Efficient bandit convex optimization: Beyond linear losses," in *Proc. Conf. Learn. Theory*, 2021, pp. 4008–4067.

[52] H. Luo, M. Zhang, and P. Zhao, "Adaptive bandit convex optimization with heterogeneous curvature," 2022, *arXiv:2202.06150*.

[53] P. Narula, P. Gutheim, D. Rolnitzky, A. Kulkarni, and B. Hartmann, "Mobileworks: A mobile crowdsourcing platform for workers at the bottom of the pyramid," in *Proc. Workshops 25h AAAI Conf. Artif. Intell.*, 2011, pp. 121–123.

[54] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[55] Y. Censor, "Pareto optimality in multiobjective problems," *Appl. Math. Optim.*, vol. 4, no. 1, pp. 41–59, 1977.

[56] P. Zhao, L. Zhang, Y. Jiang, and Z.-H. Zhou, "A simple approach for non-stationary linear bandits," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 746–755.

**Xuchuang Wang** received the BEng degree from the School of Electronic and Information Engineering, Xi'an Jiaotong University, in 2019. He is currently working toward the PhD degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, under the supervision of Prof. John. C.S. Lui. His research interests include online learning and sequential decision making.

**Hong Xie** (Member, IEEE) received the BEng degree from the School of Computer Science and Technology, University of Science and Technology of China, in 2010, and the PhD degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong (CUHK), in 2015. He is a researcher with the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences. He was a postdoctoral fellow with CUHK and NUS.

**John C.S. Lui** (Fellow, IEEE) received the PhD degree in computer science from the University of California, Los Angeles. He was a chairman with the CSE Department from 2005 to 2011. He is currently the Choh-Ming Li chair professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. He is an elected member with the IFIP WG 7.3, a croucher senior research fellow. He was a recipient of the various teaching awards. He was also a co-recipient of the best paper award in IFIP WG 7.3 Performance 2005, etc.