Group-Fair Online Allocation in Continuous Time

Semih Cayci[*]
scayci@illinois.edu

Swati Gupta[†]
swatig@gatech.edu

Atilla Eryilmaz[‡]
eryilmaz.2@osu.edu

Abstract

The theory of discrete-time online learning has been successfully applied in many problems that involve sequential decision-making under uncertainty. However, in many applications including contractual hiring in online freelancing platforms and server allocation in cloud computing systems, the outcome of each action is observed only after a random and action-dependent time. Furthermore, as a consequence of certain ethical and economic concerns, the controller may impose deadlines on the completion of each task, and require fairness across different groups in the allocation of total time budget B. In order to address these applications, we consider continuous-time online learning problem with fairness considerations, and present a novel framework based on continuous-time utility maximization. We show that this formulation recovers reward-maximizing, max-min fair and proportionally fair allocation rules across different groups as special cases. We characterize the optimal offline policy, which allocates the total time between different actions in an optimally fair way (as defined by the utility function), and impose deadlines to maximize time-efficiency. In the absence of any statistical knowledge, we propose a novel online learning algorithm based on dual ascent optimization for time averages, and prove that it achieves $\widetilde{O}(B^{-1/2})$ regret bound.

1 Introduction

With the prevalence of automated decision methods and machine learning methods, it is important to analyze the impact of learning and evaluate models not only with respect to traditional objectives such as reward or model accuracy, but also to account for the impact on individuals that interact with the system. Indeed, there are many studies highlighting algorithmic discrimination due to problems in the machine learning pipeline: imbalance in data [1], learnt representations [2, 3], choice of model proxies [4], demographic group-dependent difference in error rates of the learned models [5, 6, 7], to name a few. With rising ethical and legal concerns, addressing such issues has become urgent, specially as these impact critical societal decisions involving job opportunities and hiring. In 2014, it was estimated that 25% of the total workforce in the US was involved in some form of freelancing, and this number was predicted to grow to 40% by 2020 [8]. In reality, this percentage might be much higher, due to COVID-19 restrictions leading to increased work-from-home and changes in job opportunities [9, 10]. In online platforms however, there has been a strong evidence of bias observed in number of user reviews and user ratings[4] on completing jobs with significant correlations

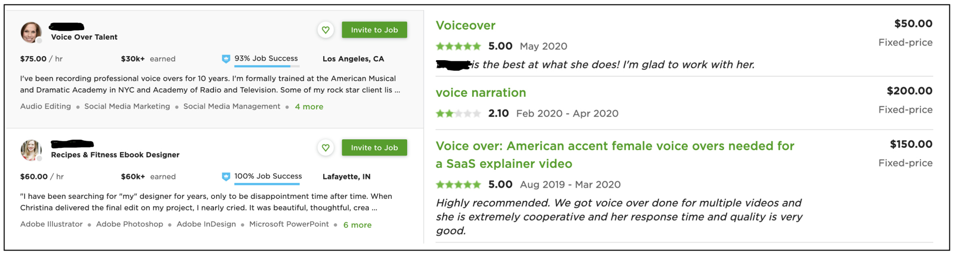
[*] Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801

[†] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332

[‡] Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210

[4] The mean (median) normalized rating score for White workers was 0.98 (1), while it is 0.97 (1) for Black workers on TaskRabbit. The mean (median) rating of White workers was found to be 3.3 (4.8), 3.0 (4.6) for Black workers, 3.3 (4.8) for Asian workers, 3.6 (4.8) for workers with a picture that does not depict a person, and 1.7 (0.0) for workers with no image on Fiverr [11].

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.



[redacted]
Voice Over Talent

$75.00 / hr $30k+ earned 93% Job Success Los Angeles, CA

I've been recording professional voice overs for 10 years. I'm formally trained at the American Musical and Dramatic Academy in NYC and Academy of Radio and Television. Some of my rock star client lis ...

Audio Editing • Social Media Marketing • Social Media Management • 4 more

[redacted]
Recipes & Fitness Ebook Designer

$60.00 / hr $60k+ earned 100% Job Success Lafayette, IN

"I have been searching for "my" designer for years, only to be disappointment time after time. When Christina delivered the final edit on my project, I nearly cried. It was beautiful, thoughtful, crea ...

Adobe Illustrator • Adobe Photoshop • Adobe InDesign • Microsoft PowerPoint • 6 more

Voiceover $50.00
★★★★★ 5.00 May 2020 Fixed-price
[redacted] is the best at what she does! I'm glad to work with her.

voice narration $200.00
★★☆☆☆ 2.10 Feb 2020 - Apr 2020 Fixed-price

Voice over: American accent female voice overs needed for a SaaS explainer video $150.00
★★★★★ 5.00 Aug 2019 - Mar 2020 Fixed-price
Highly recommended. We got voice over done for multiple videos and she is extremely cooperative and her response time and quality is very good.

Figure 1: Freelancer profiles on UPWORK with their past performance and corresponding reviews for “fixed-price" contracts. Contractors can access these profiles and allocate fixed-timed contracts with deadlines.

with race, gender, location of work and length of profiles[5] [11]. Motivated by these problems in online contractual hiring, we study a theoretical framework for sequential resource allocation to workers, where the controller (decision maker) can enforce deadlines for each task's completion. Our key contribution is to quantify impact of reward maximization in terms of equality of opportunity for jobs and develop algorithms that can achieve a meaningful trade-off between these via online utility maximization. The challenge is to maximize total reward within a given time budget, while accounting for random completion times by workers from different groups and fairness in allocation.

Formally, we consider K groups of individuals who can be hired sequentially for each task, i.e., at any point, exactly one individual can be hired. If an individual from group $k \in [K]$ is chosen for the n-th task and given a contractual deadline t by the controller, he/she generates a random *reward* of $R_{k,n}$ if the task is completed by (random) time $X_{k,n}$ within deadline t. If the task is not completed by the deadline, the reward obtained by the controller is zero and the time until the deadline is wasted (i.e., yields 0 reward for the controller). Completion times and reward distributions are assumed group-dependent and i.i.d. across tasks. The objective of the controller is to maximize utility (trade-off between total reward and fair allocation) in the offline (known distributions) and online settings (unknown distributions) under a budget constraint on time. As we will show in this paper, controlled deadlines set are essential for optimal time-efficiency under the budget constraint.

The ethical problems we are concerned with involve the rate of jobs allocated to different demographic groups and the deadlines imposed on these under reward maximization regimes [11]. Our sequential framework would also apply to other settings, for e.g., comparative clinical trials with varying follow-up durations as well as to server allocation in cloud computing where jobs are drawn from different application groups and must commit computational resources until a specific amount of time due to service level agreements (Section 2). We will often focus on the first application involving online contractual hiring, since fairness concerns are most naturally motivated in this domain.

Given a time budget constraint B and the diverse random nature of completion time and reward pairs, the main question we consider is how to decide distribution of tasks and deadlines between different groups of people. Two potential extreme allocations are: (i) *Reward-maximizing task allocation*: The controller assigns all tasks to the most rewarding group to maximize the total reward within the given time budget. The other groups do not get any chance to receive tasks. (ii) *Proportional task allocation*: The controller completely ignores the reward distributions, and attempts to give equal time share to each group. In other words, each group receives a fraction of the tasks inversely proportional to their mean completion times. There is clearly a trade-off between the reward maximization and equal time-share considerations in continuous-time sequential task allocation, and well-chosen utility functions [12] can be helpful in modeling this in a unified way. In this paper, we consider a very general class of utility functions, which recovers broadly used fairness criteria such as proportional fairness, max-min fairness, reward maximization among many others [13, 14, 12]. The controller can determine her priorities in terms of notions of fairness and model the task allocation problem by choosing the utility function accordingly.

The main contributions of this paper are summarized as follows:

1. **Incorporation of random completion time dynamics and fairness in allocation:** In discrete-time online learning models, each action is assumed to take a unit completion time, thus the

[5] Mean (median) number of reviews: for women 33 (11), 59 (15) for men on TASKRABBIT. Mean (median) number of reviews: for Black workers was found to be 65 (4), 104 (6) for White workers, 101 (8) for Asian workers, 94 (10) for non-human pictures and 18 (0) for users with no image on FIVERR [11].

random and diverse nature of task completion times, as required in many fundamental real-life applications, is ignored. In this work, we incorporate this aspect and develop a sequential learning framework in continuous time using tools from the theory of renewal processes and stochastic control. We show how controlled deadlines improve the time-efficiency in continuous-time decision processes. Moreover, this is the first work, to the best of our knowledge, that analyzes fair distribution policies in online contractual hiring.

2. **Characterization of Approximately Optimal Offline Policies:** As a consequence of the random and controlled task completion times, the optimal policy for fair resource allocation is PSPACE-hard akin to unbounded stochastic knapsack problems. For tractability in design and analysis, we propose an approximation to the optimal offline policy based on Lagrange duality and renewal theory, and prove that it is asymptotically optimal. These approximate policies allocate tasks independently with respect to a fixed probability distribution.
3. **Online learning for utility maximization:** For utility maximization in an online setting with full information feedback, we develop a novel and low-computational-complexity online learning algorithm based on dynamic stochastic optimization methods for time averages, and show that it achieves $\tilde{O}(B^{-1/2})$ regret for a time budget B. The optimal offline control policy in this paper is time-dependent, randomized and attempts to optimize time averages unlike the reward maximization problems in discrete-time problems. Despite these, the online learning algorithm we developed adapts to the randomness in completion time-reward pairs, and achieves optimal performance with vanishing regret at a fast rate.

Related Work: The problem of fair resource allocation via utility maximization has been widely considered in economics and network management [15, 16, 17, 18]. The utility maximization approach to fair resource allocation in these papers predominantly deals with discrete-time systems, therefore the randomness and diversity in task completion times is completely ignored. Furthermore, these works either assume perfect knowledge of rewards and completion times prior to decision-making, or they assume the knowledge of statistics, therefore they do not incorporate online learning. The only continuous-time utility maximization approach to fair resource allocation is [19], which assumes the knowledge of first-order statistics, thus considers an offline optimization setting. Our work utilizes the (offline) Lyapunov optimization methods proposed in [19] to develop online learning algorithms based on the notion of approximate Lyapunov drift.

Online learning under budget constraints has been considered under the scope of bandits with knapsacks [20, 21, 22]. In the classical bandits with knapsacks model, the objective is to maximize expected total reward under knapsack constraints in a stochastic setting. In [23], an interrupt mechanism is employed to incorporate the continuous-time dynamics into the budget-constrained online learning model. Note that these works focus solely on reward maximization, therefore do not address the fair resource allocation problem. The bandits with knapsacks setting was extended to concave rewards and convex constraints in [24], which assumes bounded cost and reward, and the deadline mechanism is not involved in decision-making, thus optimal time-efficiency in continuous time is not achieved. Our paper deviates from this line of work as it proposes a versatile and comprehensive framework for fairness, and incorporates continuous-time dynamics into the decision-making for time-efficiency. We include an extended discussion of related work in Appendix A.

2 Online Learning Framework for Group Fairness

We consider the sequential and fair allocation of tasks to individuals from different groups, whose completion times and rewards randomly vary. This goal differs significantly from traditional online learning models that aim to maximize the expected total reward with unit completion times. Under this traditional setting, the controller's goal is to find and persistently select the reward-maximizing groups to allocate its tasks. As a consequence, the reward-maximization objective leads to the starvation of suboptimal groups, which causes unfairness amongst the groups with different statistical characteristics. Next, we provide a few motivating examples with group fairness requirements:

- **Contractual Hiring in Online Freelancing Platforms:** Online freelancing sites like UPWORK host contractual workers (freelancers) that can be hired by "contractors" who require specific tasks to be completed. Each freelancer has a profile and performance on past tasks that can be learned by the contractors via ratings and reviews (see, typical profile in Figure 1). Fixed-timed contracts are popular on UPWORK, wherein contractors enforce a deadline by which the task must be completed otherwise the contract is terminated (i.e., there is no payment). Contractors can browse profiles and post a job to a selected set of freelancers with a deadline. However, there is a large literature documenting bias in online rating systems, which in turn impact job opportunities disparately [11, 25, 26], thus making it critical to develop theory of online learning for such settings.

- **Server Allocation in Cloud Computing:** An important application of our framework is online learning for fair resource allocation in cloud computing systems. In a very basic setting, a single server is sequentially allocated to tasks from one of K user groups, which exhibit similar execution time statistics and priority levels within each group. In many practical scenarios, the execution time of a task is unknown at the time decision [27, 28], and exhibits a power-law behavior [29], which necessitates a deadline mechanism for optimal time-efficiency [23]. In this setting, the objective of the controller is to allocate the server in an optimally fair way across the groups in a given time interval $[0, B]$, depending on the completion time statistics and priority levels. Our work proposes a versatile framework to model fairness for this problem based on the concept of continuous-time utility maximization, and develops online learning algorithms to achieve the optimal performance with low regret in the absence of any statistical knowledge.

More examples can be found in other domains, including multi-user wireless communication over fading channels (e.g., see [23]), comparative clinical trials with optimal follow-up duration (e.g., see [30, 31]), whereby the goal is to fairly share the limited resources between groups of users.

Motivated by these examples, next we introduce an online learning framework that expands the traditional setting substantially to incorporate group fairness characteristics into its formulation. Suppose that there are $K \geq 1$ groups of individuals that are available for serving tasks, and the controller assigns tasks sequentially among these K groups. If an individual from group k is chosen for the n-th task, he/she takes $X_{k,n}$ units of *completion time*, and a *reward* of $\overline{R}_{k,n}$ is obtained upon successful completion, where $X_{k,n}$ and $\overline{R}_{k,n}$ are positive random variables. We assume that individuals within a group exhibit statistical similarities as a consequence of their common background, therefore we assume that the process $(X_{k,n}, \overline{R}_{k,n})$ is independent and identically distributed (iid) over n. In order to model the possibility of highly different skill sets for individuals within a group, we assume that the completion time $X_{k,n}$ is random, and can be potentially heavy-tailed. Note that the completion time $X_{k,n}$ and reward $\overline{R}_{k,n}$ can be correlated, e.g., in the server allocation example, the completion time $X_{k,n}$ and size $\overline{R}_{k,n}$ of a task are positively correlated [32].

Before the n-th task begins, the controller makes two decisions: the group $G_n \in [K]$ of the individual that will be assigned the task, and a deadline $T_n \in \mathbb{T}$, where $\mathbb{T} \subset \mathbb{R}_+$ is the set of all possible deadlines. Since $(X_{k,n}, \overline{R}_{k,n})$ is independent and identically distributed over n and unknown at the time of decision, i.e., each individual is statistically symmetric, we do not consider the assignment of tasks to individuals within a group, thus the controller only makes a choice for the group in task allocation. If the task is not completed by the selected deadline $t \in \mathbb{T}$, the task is interrupted without collecting any reward, therefore we denote the reward obtained t time units after the initiation by $R_{k,n}(t) = \overline{R}_{k,n}\mathbb{I}\{X_{k,n} \leq t\} \leq R_{max}$ for some constant $R_{max} < \infty$. In many applications, deadlines are chosen within a discrete set (e.g., days/months in contractual hiring or time-slots in server allocation), thus we assume a finite decision set $\mathbb{T} = \{t_1, t_2, \ldots, t_L\}$ with $t_l < \infty$ for all l in this paper. The sequential task allocation continues until a given time budget $B > 0$ is exceeded, therefore, the completion time of a task is as important as the reward.

To describe this process mathematically, let $\mathcal{H}_{k,n-1}$ denote the available feedback for group k, and $\mathcal{H}_{n-1} = \cup_{k\in[K]}\mathcal{H}_{k,n-1}$ denote the history before making a decision for task n. For a given time budget $B > 0$, a *causal policy* $\pi = \{\pi_1, \pi_2, \ldots\}$ sequentially makes two decisions $\pi_n = (G_n, T_n) \in [K] \times \mathbb{T}$ for each task n based on the history $\mathcal{H}_{k-1}$, where G_n is the chosen group and T_n is the assigned deadline. Under a policy π, the number of initiated tasks is the following *first-passage time*:

$$N^\pi(B) = \inf\Big\{n : \sum_{i=1}^{n} \min\{X_{G_i,i}, T_i\} > B\Big\}, \tag{1}$$

which is a random and controlled stopping time. Moreover, the *reward rate* of any user type k is:

$$\overline{r}_k^\pi(B) = \mathbb{E}\Big[\frac{1}{B}\sum_{n=1}^{N_\pi(B)} \mathbb{I}\{G_n = k\}R_{k,n}(T_n)\Big], \text{ under policy } \pi. \tag{2}$$

If $R_{k,n}(t) = \mathbb{I}\{X_{k,n} \leq t\}$, i.e., each task completion yields a unit reward, then $\overline{r}_k^\pi(B)$ simply denotes the task completion rate (i.e., throughput) of group k individuals in the time interval $[0, B]$.

Note that designing strategies that aim to maximize the total reward rate in (2) will lead to the persistent selection of the group with the highest reward rate at the cost of starvation of all the rest (see [23]). In order to address group fairness considerations, we propose a continuous-time online learning framework based on the utility maximization concept that is used effectively in the fair resource allocation domain (e.g., see [16]). Specifically, for a given continuously-differentiable, concave and monotonically increasing utility function $U_k : \mathbb{R} \to \mathbb{R}$, we let the utility of group k under a policy π be given by $U_k\big(\overline{r}_k^\pi(B)\big)$. Then, the total utility under a policy π is defined as:

$$U^\pi(B) = \sum_{k=1}^{K} U_k\big(\overline{r}_k^\pi(B)\big), \text{ for time interval } [0, B].$$

For a given time budget $B > 0$, the optimum total utility over a class of policies Π, and the regret for a given policy $\pi \in \Pi$ are, respectively:

$$\mathtt{OPT}_\Pi(B) = \max_{\pi\in\Pi} \sum_{k=1}^{K} U_k\big(\overline{r}_k^\pi(B)\big) \quad \text{and} \quad \mathtt{REG}_\Pi^\pi(B) \quad = \mathtt{OPT}_\Pi(B) - U^\pi(B). \tag{3}$$

Note that, due to the monotonically increasing and concave nature of utility functions, allocating the tasks always to the most rewarding group is not a good choice, because the same amount of time could yield a higher utility for another group because of the diminishing return property of concave functions. Thus, each given set of utility functions $\{U_k : k \in [K]\}$ defines a fairness concept. A particularly important class of utility functions is α-fair class, given next.

Definition 1 (α-Fair Allocation). *For any given $\alpha > 0$ and weight $w_k > 0$, let $U_k(x) = w_k \frac{x^{1-\alpha}}{1-\alpha}$, for all k. Resource allocation by using these utility functions is called α-fair resource allocation.*

This class is attractive since it includes as special cases proportional fairness, minimum potential delay fairness, reward maximization and max-min fairness [12].

3 Approximation of the Optimal Offline Policy

Note that a simpler version of the sequential maximization problem in (3) with linear utility functions over all causal policies is called an unbounded knapsack problem, and it is PSPACE-hard even in the case of known statistics [33, 20]. Therefore, the optimal causal policy for the problem in (3) has a very high computational complexity even in the offline setting, which makes it intractable for online learning. For tractability in design and analysis, we consider a class of simple policies that allocate tasks in an i.i.d. randomized way according to a fixed probability distribution over groups, and show its efficiency in this section.

Definition 2 (Stationary Randomized Policies). *Let P be a fixed probability distribution over $[K] \times \mathbb{T}$. A stationary randomized policy (SRP) $\pi = \pi(P)$ makes a randomized decision independently according to P for every task until the time budget B is depleted. In other words, under the SRP $\pi(P)$, we have $\mathbb{P}\big(\pi_n = (k,t)\big) = P(k,t)$, $\forall n \leq N_\pi(B)$, for all $(k,t) \in [K] \times \mathbb{T}$. We denote the class of all stationary randomized policies as Π_S.*

Proposition 1 (Asymptotic optimality of SRP). *There exists a probability distribution $P^\star$ such that the stationary randomized policy $\pi(P^\star)$ is asymptotically optimal over all causal policies as $B \to \infty$.*

The proof of Proposition 1 can be found in Appendix B. In the following, we characterize the total utility under $\pi(P)$ by providing tight bounds.

Proposition 2. *Let P be any given probability distribution over $[K] \times \mathbb{T}$. Then, the reward per unit time for group k under the stationary randomized policy $\pi(P)$ is as follows:*

$$\rho_k(P) = \frac{\sum_{t\in\mathbb{T}} P(k,t)\mathbb{E}[R_{k,1}(t)]}{\sum_{(i,t)\in[K]\times\mathbb{T}} P(i,t)\mathbb{E}[\min\{X_{i,1}, t\}]}, \forall k \in [K].$$

Consequently, the total utility under the stationary randomized policy $\pi(P)$ is bounded as follows:

$$\sum_{k\in[K]} U_k\Big(\rho_k(P)\Big) \le \sum_{k\in[K]} U_k\big(\overline{r}_k^{\pi(P)}(B)\big) \le \sum_{k\in[K]} U_k\Big(\rho_k(P)\Big) + O\Big(\frac{1}{B}\Big).$$

We include the complete proof of Proposition 2 in Appendix B. The key idea is that under an SRP, the total reward of a group k is a regenerative process. Then, by using the theory of stopped random walks for regenerative processes, the reward per unit time under $\pi(P)$ is found as $\rho_k(P)$, and the upper bound for the total utility is found by using Lorden's inequality [34] and concavity of U_k.

Proposition 2 emphasizes the significance of the reward per unit time $\rho_k(P)$. In conjunction with Proposition 1, this suggests that using a probability distribution that maximizes the limiting total utility would be an effective offline approximation.

Definition 3 (Optimal Stationary Randomized Policy)**.** *Let $P^\star$ be a probability distribution defined as $P^\star \in \arg\max_P \sum_{k\in[K]} U_k\Big(\rho_k(P)\Big)$. Then, the optimal SRP $\pi^\star$ makes a selection independently for every task according to $P^\star$: $\mathbb{P}\Big(\pi_n^\star = (k,t)\Big) = P^\star(k,t)$ for all $(k,t) \in [K]\times\mathbb{T}$ and $n \le N_\pi(B)$.*

An interesting question regarding $P^\star$ is the choice of deadline policy for each group. The following proposition characterizes the optimal deadline policy under $\pi^\star$, and yields a significant simplification in finding the optimal policy by reducing the size of the search space.

Proposition 3 (Optimal Deadline Policy)**.** *For any k, the optimal probability distribution $P^\star$ makes a deterministic deadline decision for group k, that is, $|\{t \in \mathbb{T} : P^\star(k,t) > 0\}| \le 1$. For any k, we denote $t_k^* \in \mathbb{T}$ as the (unique) optimal deadline for group k such that $P^\star(k, t_k^*) > 0$.*

The detailed proof of Prop. 3 can be found in Appendix C. As we will see later, we can explicitly characterize the optimal deadline for a broad class of utility functions used for the so-called α-fair allocations. In the following, we use Prop. 2 to characterize the performance of the optimal SRP.

Proposition 4 (Optimal Total Utility)**.** *For any group k, let $t_k^* \in \mathbb{T}$ be the (unique) optimal deadline by Prop. 3; $r_k^* = \mathbb{E}[R_{k,1}(t_k^*)]/\mathbb{E}[\min\{X_{k,1}, t_k^*\}]$ be the reward per processing time for group k; and*

$$\varphi_k = \frac{P^\star(k,t_k^*)\cdot\mathbb{E}[\min\{X_{k,1},t_k^*\}]}{\sum_{j\in[K]} P^\star(j,t_j^*)\cdot\mathbb{E}[\min\{X_{j,1},t_j^*\}]}, \tag{4}$$

be the fraction of time budget allocated to group k under $\pi(P^\star)$. Then, for any SRP $\pi(P)$, the total utility is bounded as $\sum_k U_k\Big(\rho_k(P)\Big) \le \sum_k U_k\Big((U_k')^{-1}\big(\frac{\lambda}{r_k^*}\big)\Big)$, where the upper bound is achieved by the probability distribution that satisfies $\varphi_k = \frac{1}{r_k^*}(U_k')^{-1}\big(\frac{\lambda}{r_k^*}\big)$ for λ such that $\sum_k \varphi_k = 1$.

The proof of Proposition 4 follows from Lagrange duality and Prop. 3, and can be found in Appendix D. Note that the above analysis is very general in the sense that it holds for any set of utility functions $\{U_k : k \in [K]\}$ that are continuously differentiable and concave. In the following, we apply the results to the class of α-fair allocations (cf. Definition 1) and discuss their implications.

Proposition 5 (α-Fair Resource Allocation in Continuous Time)**.** *For any group k, the optimal deadline optimizes the reward per processing time:*

$$t_k^* = \arg\max_{t\in\mathbb{T}} \frac{\mathbb{E}[R_{k,1}(t)]}{\mathbb{E}[\min\{X_{k,1},t\}]}.$$

Let $r_k^* = \max_{t\in\mathbb{T}} \frac{\mathbb{E}[R_{k,1}(t)]}{\mathbb{E}[\min\{X_{k,1},t\}]}$ be the reward per processing time and $\mu_k = \mathbb{E}[\min\{X_{k,1}, t_k^*\}]$ be the mean processing time for group k under the optimal deadline policy. Then, for any $\alpha > 0$, we have the following results for α-fair utility functions:

$$\max_P U^{\pi(P)}(B) = \frac{1}{1-\alpha}\Big(\sum_{k\in[K]} (r_k^*)^{\frac{1}{\alpha}-1} w_k^{\frac{1}{\alpha}}\Big)^{\alpha}, \tag{5}$$

where the optimum probability distribution P_k^* and the optimum fraction of time budget φ_k allocated to group k are, respectively, given by:

$$P^\star(k,t) = \mathbb{I}\{t = t_k^*\} \frac{w_k^{\frac{1}{\alpha}} (r_k^*)^{\frac{1}{\alpha}-1}/\mu_k}{\sum_{j\in[K]} w_j^{\frac{1}{\alpha}} (r_j^*)^{\frac{1}{\alpha}-1}/\mu_j},$$

$$\varphi_k = \frac{(r_k^*)^{\frac{1}{\alpha}-1} w_k^{\frac{1}{\alpha}}}{\sum_{j\in[K]} (r_j^*)^{\frac{1}{\alpha}-1} w_j^{\frac{1}{\alpha}}},$$

for all $(k,t) \in [K] \times \mathbb{T}$.

Proposition 5 characterizes the optimal stationary randomized policy for a wide class of utility functions, and implies that the optimal deadline for a group is chosen to maximize the reward per processing time for that group in this setting.

To gain a clear understanding of the notion of α-fairness, we consider the following special cases.

Corollary 1. *For any given set of parameters $\{w_k > 0 : k \in [K]\}$, we have the following results for continuous-time α-fair resource allocation problem for various $\alpha > 0$ values.*

(i) ***Proportional fairness:*** *In this case, we have $\lim_{\alpha\to 1} U_k(x) = w_k \log(x)$ for all k. Let $\mu_k = \mathbb{E}[\min\{X_{k,1}, t_k^*\}]$ be the mean processing time for group k. Then, the optimum utility is achieved by the probability distribution $P^\star(k,t) = \mathbb{I}\{t = t_k^*\} \frac{w_k/\mu_k}{\sum_{j\in[K]} w_j/\mu_j}$, $(k,t) \in [K] \times \mathbb{T}$, thus we have $\varphi_k = \frac{w_k}{\sum_{j\in[K]} w_j}$ for all k, which implies the time budget B is allocated to a group k proportional to its weight w_k, and the optimal total utility is $\texttt{OPT}_{\Pi_S}(B) = \sum_k \log\left(\frac{r_k^* w_k}{\sum_{k'\in[K]} w_{k'}}\right) + O(\frac{1}{B})$.*

(ii) ***Reward maximization:*** *If $\alpha = 0$, we have $U_k(x) = \omega_k x$ for all k. Let $k^* = \arg\max_{k\in[K]}\ w_k r_k^*$ be the group with highest weighted reward rate. Then, the optimal probability distribution is $P^\star(k,t) = \mathbb{I}\{k = k^*,\ t = t_k^*\}$, for all (k,t). Thus, $\texttt{OPT}_{\Pi_S}(B) = \max_{k\in[K]} w_k r_k^* + O(1/B)$.*

Remark 1. Note that optimal deadline t_k^* for any group k is chosen so as to maximize the reward per processing time of group k. Under proportional fairness ($\alpha \to 1$), the controller distributes the time budget proportional to group weights, i.e., $\varphi_k = w_k / \sum_j w_k$, which reduces to equal time-sharing under uniform weights. To achieve this, the controller allocates tasks with probability inversely proportional to the mean processing time μ_k. Under reward maximization ($\alpha = 0$), the controller allocates the entire time budget B to a single group that yields the highest reward per processing time to maximize the expected total reward, i.e., $\varphi_k = \mathbb{I}\{k = k^*\}$. As such, the trade-off between reward maximization and equal (i.e., reward-insensitive) time-sharing is modeled by α-fairness for any $\alpha \in [0,1)$. Further, the α-fair utility maximization framework includes max-min fairness ($\alpha \to \infty$) and minimum potential delay fairness ($\alpha = 2$) as subcases.

4 Online Learning for Utility Maximization (`OLUM`)

In the previous section, we provided key results on the asymptotically optimal approximations to the offline utility maximization problem. In this section, we will build on these to attack the online learning problem for continuous-time fair allocation. In particular, we will propose a novel online learning algorithm for the fair resource allocation problem based on renewal theory, Lyapunov optimization (see [19, 35]) and PAC-bandits, and show that it achieves $\widetilde{O}(B^{-1/2})$ regret.

Feedback model: We assume a delayed full-information feedback model where the completion time and reward of all groups for task n are revealed to the controller at stage $n + \tau$ for some delay $\tau \geq 1$.

This assumption holds approximately for our target applications. In freelancing platforms, there are often multiple contractors that hire freelancers for various tasks. It is often possible to get full information on various freelancers due to employment by other companies and their reviews can serve as the feedback for the controller. Competitions hosting websites like TOPCODER have also recently been catering to businesses who need fast-prototyping using freelancers. In their business model, a controller might invest in a few topcoders at a time, however, she can potentially get access to updated rankings (quality and time to complete tasks) via topcoder competitions over time. In server applications such as Amazon AWS and Microsoft Azure as well, although a controller might be optimizing operations on a local set of servers, they can request task performance data from a centralized server or a scheduler after a delay in time [36]. This feedback model already presents with technical challenges due to random completion times, as we discuss next.

In order to design the online learning algorithm, let us define, for any $(k, t) \in [K] \times \mathbb{T}$, the empirical estimates of the mean completion time and reward after n stages, respectively, as

$$\widehat{\mu}_{k,n}(t) = \frac{1}{n}\sum_{i=1}^{n} \min\{t, X_{k,i}\}, \qquad \text{and} \quad \widehat{\theta}_{k,n}(t) = \frac{1}{n}\sum_{i=1}^{n} R_{k,i}(t).$$

Definition 4 (`OLUM` Algorithm). *For any k, let $Q_{k,0} = 1$ and $Q_{k,i}$ be defined recursively as follows:*

$$Q_{k,i+1} = \Big(Q_{k,i} + \gamma_k(i)\min\{X_{G_i,i}, T_i\} - R_{k,i}(T_i)\mathbb{I}\{G_i = k\}\Big)^+, \qquad i > 0 \tag{6}$$

where the auxiliary variable $\gamma_k(i) = \big(U_k'\big)^{-1}\Big(Q_{k,i}/V\Big)$, where $V > 0$ is a design choice. Then, for the task n, the `OLUM` Algorithm, denoted by $\pi^{\texttt{OLUM}}$, makes the following decision:

$$(G_n, T_n) \in \underset{(k,t)\in[K]\times\mathbb{T}}{\arg\max} \frac{\widehat{\theta}_{k,n-\tau}(t)Q_{k,n}}{\widehat{\mu}_{k,n-\tau}(t)}.$$

Upon observing the corresponding feedback, the controller updates $Q_{k,n+1}$ via (6).

Interpretation: The `OLUM` Algorithm aims to maximize the time-average reward weighted with $Q_{k,n}$ at each round. Note that for any $k \in [K]$, if the sequence $Q_{k,n}$ gets very big, then its reward rate is much smaller than the optimal value, thus the controller tends to select that group. In other words, the magnitude of $Q_{k,n}$ is a measure of the unfairness that group k has endured by stage n. The algorithm is designed so as to balance the weights $Q_{k,n}$ to maximize the total utility.

In the following theorem, we prove regret bounds for the `OLUM` Algorithm.

Theorem 1 (Regret bounds for `OLUM`). *For any $V > 0$ and constant delay τ, the regret under $\pi^{\texttt{OLUM}}$ is bounded as $\texttt{REG}_{\Pi_S}\pi^{\texttt{OLUM}}(B) = O\Big(\sqrt{\frac{\log(B)}{B}} + \frac{V}{B} + \frac{1}{V}\Big)$. By choosing $V = \Theta(\sqrt{B/\log(B)})$, we obtain $\texttt{REG}_{\Pi_S}\pi^{\texttt{OLUM}}(B) = O(\sqrt{\log(B)/B}) = \widetilde{O}(1/\sqrt{B})$.*

The proof is based on renewal theory, Lyapunov optimization theory and PAC-type bounds, and can be found in Appendix E.

5 Simulations

We implemented the `OLUM` Algorithm on a fair resource allocation problem with $K = 2$ groups. In the application domains that we considered in Section 2, the task completion times naturally follow a power-law distribution. For the contractual online hiring setting, creativity of individuals has been shown to follow a Pareto(1, γ) distribution with exponent $\gamma > 1$, where γ is dependent on the field of expertise [37]. Motivated by this application, we consider the following group statistics:

- **Group 1:** $X_{k,n} \sim$ Pareto$(1, 1.2)$ and $R_{k,n}(t) = X_{k,n}^{0.6} \cdot \mathbb{I}\{X_{k,n} \le t\}$
- **Group 2:** $X_{k,n} \sim$ Pareto$(1, 1.4)$ and $R_{k,n}(t) = X_{k,n}^{0.2} \cdot \mathbb{I}\{X_{k,n} \le t\}$

The reward per processing time as a function of the deadline is shown in Figure 2. For this setting, we implemented the `OLUM` Algorithm with parameter $V = 20$, and considered α-fair resource allocation problems with various α values. In Figure 2, we present the simulation results for φ_2, i.e., the average fraction of time budget B allocated to Group-2 individuals, under the `OLUM` Algorithm. For these experiments, we chose $w_k = 1$ for $k = 1, 2$ and ran the `OLUM` Algorithm for 1000 trials for each set. Note that the optimal reward per processing time of Group-1 individuals is higher than that of Group-2 individuals, thus Group-1 is chosen for reward maximization. Under proportional fairness, the time budget is equally distributed between Group-1 and Group-2 individuals. We observe from Figure 2 that the `OLUM` Algorithm converges to the optimal operating points very fast, which verifies the theoretical results we presented.

In the second example, we consider a fair server allocation problem with $K = 5$ user groups. In processing systems with shared servers, empirical studies indicate that the distribution of job

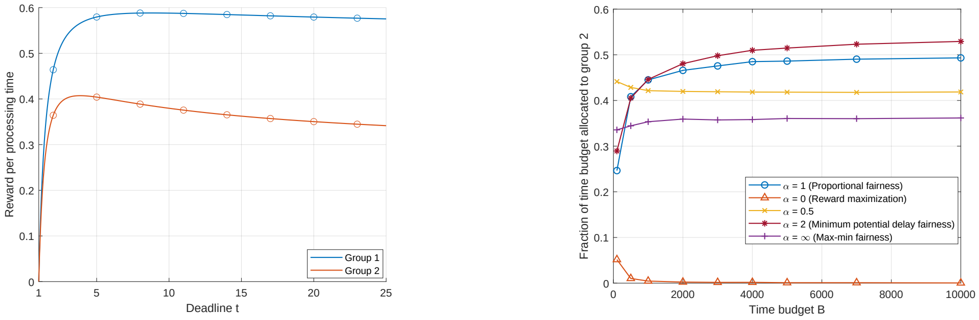


Figure 2: (Left) Reward per processing time for each group. (Right) Fraction of time budget assigned to Group-2 individuals under the OLUM Algorithm for various fairness criteria.

execution times (i.e., file sizes) can be accurately approximated by Pareto(1, γ) distribution with $\gamma \in (0,2)$ [38, 39, 40]. Thus, we consider Pareto distributed completion times for each group with varying exponents: $X_{k,n} \sim Pareto(1, \alpha_k)$ for $\alpha = (1.25, 1.50, 1.75, 2.00, 2.25)$. The reward of group k under a deadline $t \in \mathbb{T}$ is $R_{k,n}(t) = \rho_k \cdot X_{k,n}^{\beta_k} \cdot \mathbb{I}\{X_{k,n} \leq t\} \leq \rho_k t_L^{\beta_k}$ where β_k measures the correlation. For this example, we consider $\beta = (0,\ 0.125,\ 0.25,\ 0.375,\ 0.5)$ and $\rho = (3.0,\ 2.0,\ 3.0,\ 1.0,\ 1.44)$. Reward per processing times as a function of deadline are shown in Fig. 3, where the deadline choices $t \in \mathbb{T}$ are marked by squares. We observe that the optimal deadline critically depends on the tail exponent and correlation between completion time and reward [23]. The performance of the OLUM Algorithm for proportionally fair allocation of the server (i.e., $U_k(x) = \log(x),\ k \in [K]$) is shown in Fig. 3 for various choices of the parameter V. In Theorem 1, we showed that $\widetilde{O}(1/\sqrt{B})$ regret is achieved when the parameter V is chosen as $V = \Theta(\sqrt{B/\log(B)})$ for time budget B. From Fig. 3, we verify that this scaling of the parameter V is necessary for good practical performance.

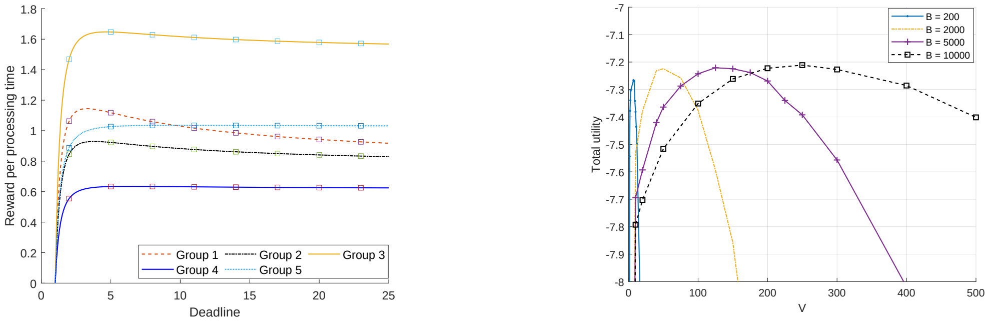


Figure 3: (Left) Reward per processing time for each group. (Right) Total utility for various time budgets and design parameter V choices.

6 Conclusion

In this paper, we proposed a versatile and comprehensive framework for continuous-time online resource allocation with fairness considerations, and proposed a no-regret learning algorithm for this problem in a delayed full-information feedback model. Note that although the full-information feedback is available in many application scenarios, there are cases in which the controller does not have an access to full feedback, thus a mechanism that incorporates bandit feedback is required. The online learning framework introduced in this paper can be extended to bandit feedback. One way to achieve this might be to replace the empirical estimates with upper confidence bounds in the OLUM Algorithm, which makes the analysis even more complicated. We leave the design and analysis of bandit algorithms in this setting as a future work.

Broader Impact

Our work develops the theory of fair online learning, specifically analyzing the impact of reward-maximizing allocation policies on opportunities for different groups of people. Our proposal analyzes the trade-offs across various allocation policies (ranging from profit maximizing to equal opportunity for all), thus highlighting the choice of objectives that the controllers should carefully consider. This work does not have any foreseeable negative ethical or societal impact.

Acknowledgments and Disclosure of Funding

This work was completed when S. Cayci was a PhD candidate and research assistant at The Ohio State University, Department of Electrical and Computer Engineering. This research was supported in part by: NSF grants: CNS-NeTS-1717045, CMMI-SMOR-1562065, CNS-ICN-WEN-1719371, CNS-SpecEES-1824337, CNS-NeTS-2007231; ONR Grant N00014-19-1-2621, and the DTRA grant: HDTRA1-18-1-0050. S. Gupta would like to gratefully acknowledge support from the NSF grant CRII 1850182.

References

[1] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang, "Men also like shopping: Reducing gender bias amplification using corpus-level constraints," arXiv preprint arXiv:1707.09457, 2017.

[2] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in Advances in Neural Information Processing Systems, 2016, pp. 4349–4357.

[3] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," Science, vol. 356, no. 6334, pp. 183–186, 2017.

[4] K. Lum and W. Isaac, "To predict and serve?" Significance, vol. 13, no. 5, pp. 14–19, 2016.

[5] A. L. Washington, "How to argue with an algorithm: Lessons from the compas-propublica debate," Colo. Tech. LJ, vol. 17, p. 131, 2018.

[6] J. Kleinberg, "Inherent trade-offs in algorithmic fairness," in Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems, 2018, pp. 40–40.

[7] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," Big data, vol. 5, no. 2, pp. 153–163, 2017.

[8] G. Laumeister, "The next big thing in e-commerce: Online labor marketplaces," Forbes (Online), 2014.

[9] H. Torry, "Coronavirus pandemic deepens labor divide between online, offline workers," Wall Street Journal, 2020.

[10] A. Teeley, "There are 57 million u.s. independent professionals — upwork wants them all to succeed," Built In Chicago, 2020.

[11] A. Hannák, C. Wagner, D. Garcia, A. Mislove, M. Strohmaier, and C. Wilson, "Bias in online freelance marketplaces: Evidence from taskrabbit and fiverr," in Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, 2017, pp. 1914–1933.

[12] R. Srikant and L. Ying, Communication networks: an optimization, control, and stochastic networks perspective. Cambridge University Press, 2013.

[13] D. Bertsimas, V. F. Farias, and N. Trichakis, "On the efficiency-fairness trade-off," Management Science, vol. 58, no. 12, pp. 2234–2250, 2012.

[14] K. Jain and V. V. Vazirani, "Eisenberg-gale markets: Algorithms and structural properties," in Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, 2007, pp. 364–373.

[15] D. P. Palomar and M. Chiang, “A tutorial on decomposition methods for network utility maximization,” IEEE Journal on Selected Areas in Communications, vol. 24, no. 8, pp. 1439–1451, 2006.

[16] A. Eryilmaz and R. Srikant, “Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control,” IEEE/ACM transactions on networking, vol. 15, no. 6, pp. 1333–1344, 2007.

[17] H. J. Kushner and P. A. Whiting, “Convergence of proportional-fair sharing algorithms under general conditions,” IEEE Transactions on Wireless Communications, vol. 3, no. 4, pp. 1250–1259, 2004.

[18] D. Kahneman and R. H. Thaler, “Anomalies: Utility maximization and experienced utility,” Journal of economic perspectives, vol. 20, no. 1, pp. 221–234, 2006.

[19] M. J. Neely, “Dynamic optimization and learning for renewal systems,” IEEE Transactions on Automatic Control, vol. 58, no. 1, pp. 32–46, 2012.

[20] A. Badanidiyuru, R. Kleinberg, and A. Slivkins, “Bandits with knapsacks,” Journal of the ACM (JACM), vol. 65, no. 3, pp. 1–55, 2018.

[21] L. Tran-Thanh, A. Chapman, A. Rogers, and N. R. Jennings, “Knapsack based optimal policies for budget–limited multi–armed bandits,” in Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.

[22] A. Slivkins, “Introduction to multi-armed bandits,” arXiv preprint arXiv:1904.07272, 2019.

[23] S. Cayci, A. Eryilmaz, and R. Srikant, “Learning to control renewal processes with bandit feedback,” Proceedings of the ACM on Measurement and Analysis of Computing Systems, vol. 3, no. 2, pp. 1–32, 2019.

[24] S. Agrawal and N. R. Devanur, “Bandits with concave rewards and convex knapsacks,” in Proceedings of the fifteenth ACM conference on Economics and computation, 2014, pp. 989–1006.

[25] A. Rosenblat, K. E. Levy, S. Barocas, and T. Hwang, “Discriminating tastes: Customer ratings as vehicles for bias,” Available at SSRN 2858946, 2016.

[26] A. Chakraborty, A. Hannak, A. J. Biega, and K. P. Gummadi, “Fair sharing for sharing economy platforms,” 2017.

[27] M. Harchol-Balter, “Task assignment with unknown duration,” in Proceedings 20th IEEE International Conference on Distributed Computing Systems. IEEE, 2000, pp. 214–224.

[28] R. Motwani, S. Phillips, and E. Torng, “Nonclairvoyant scheduling,” Theoretical computer science, vol. 130, no. 1, pp. 17–47, 1994.

[29] M. Harchol-Balter and A. B. Downey, “Exploiting process lifetime distributions for dynamic load balancing,” ACM Transactions on Computer Systems (TOCS), vol. 15, no. 3, pp. 253–285, 1997.

[30] K. Kim and A. A. Tsiatis, “Study duration for clinical trials with survival response and early stopping rule,” Biometrics, pp. 81–92, 1990.

[31] P. F. Thall, R. Simon, and S. S. Ellenberg, “Two-stage selection and testing designs for comparative clinical trials,” Biometrika, vol. 75, no. 2, pp. 303–310, 1988.

[32] P. R. Jelenković and J. Tan, “Characterizing heavy-tailed distributions induced by retransmissions,” Advances in Applied Probability, vol. 45, no. 1, pp. 106–138, 2013.

[33] C. H. Papadimitriou and J. N. Tsitsiklis, “The complexity of optimal queuing network control,” Mathematics of Operations Research, vol. 24, no. 2, pp. 293–305, 1999.

[34] S. Asmussen, Applied probability and queues. Springer Science & Business Media, 2008, vol. 51.

[35] M. Neely, Stochastic network optimization with application to communication and queueing systems. Morgan & Claypool Publishers, 2010.

[36] R. Zabolotnyi, P. Leitner, and S. Dustdar, “Profiling-based task scheduling for factory-worker applications in infrastructure-as-a-service clouds,” in 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications. IEEE, 2014, pp. 119–126.

[37] J. Kleinberg and M. Raghavan, "Selection problems in the presence of implicit bias," arXiv preprint arXiv:1801.03533, 2018.

[38] M. Harchol-Balter, "The effect of heavy-tailed job size distributions on computer system design." in Proc. of ASA-IMS Conf. on Applications of Heavy Tailed Distributions in Economics, Engineering and Statistics, 1999.

[39] W. J. Reed and B. D. Hughes, "From gene families and genera to incomes and internet file sizes: Why power laws are so common in nature," Physical Review E, vol. 66, no. 6, p. 067103, 2002.

[40] W. Gong, Y. Liu, V. Misra, and D. Towsley, "On the tails of web file size distributions," in Proceedings of the annual allerton conference on communication control and computing, vol. 39, no. 1. The University; 1998, 2001, pp. 192–201.

[41] J. F. Nash Jr, "The bargaining problem," Econometrica: Journal of the Econometric Society, pp. 155–162, 1950.

[42] J. W. Pratt, "Risk aversion in the small and in the large," in Uncertainty in Economics. Elsevier, 1978, pp. 59–79.

[43] N. Nisan and A. Ronen, "Computationally feasible vcg mechanisms," Journal of Artificial Intelligence Research, vol. 29, pp. 19–47, 2007.

[44] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," IEEE/ACM Transactions on networking, vol. 8, no. 5, pp. 556–567, 2000.

[45] F. Kelly, "Charging and rate control for elastic traffic," European transactions on Telecommunications, vol. 8, no. 1, pp. 33–37, 1997.

[46] L. Tassiulas and A. Ephremides, "Jointly optimal routing and scheduling in packet ratio networks," IEEE Transactions on Information Theory, vol. 38, no. 1, pp. 165–168, 1992.

[47] ——, "Dynamic server allocation to parallel queues with randomly varying connectivity," IEEE Transactions on Information Theory, vol. 39, no. 2, pp. 466–478, 1993.

[48] M. J. Neely, "A lyapunov optimization approach to repeated stochastic games," in 2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 2013, pp. 1082–1089.

[49] S. Agrawal and N. Devanur, "Linear contextual bandits with knapsacks," in Advances in Neural Information Processing Systems, 2016, pp. 3450–3458.

[50] K. A. Sankararaman and A. Slivkins, "Combinatorial semi-bandits with knapsacks," arXiv preprint arXiv:1705.08110, 2017.

[51] A. Gut, Stopped random walks. Springer, 2009.

[52] S. Cayci, A. Eryilmaz, and R. Srikant, "Budget-constrained bandits over general cost and reward distributions," arXiv preprint arXiv:2003.00365, 2020.

[53] M. J. Wainwright, High-dimensional statistics: A non-asymptotic viewpoint. Cambridge University Press, 2019, vol. 48.

[54] B. Hajek, "Hitting-time and occupation-time bounds implied by drift analysis with applications," Advances in Applied probability, vol. 14, no. 3, pp. 502–525, 1982.

A Related Work

Fair resource allocation via utility maximization has been widely studied in economics [41, 42, 18], mechanism design [43], network management [15, 44, 16, 12, 45] among many other fields. Particularly, logarithmic utility maximization was introduced in [41] for the "Nash bargaining solution" to a bargaining game among multiple players over the allocation of a shared resource, and it was used in the management of communication networks in [44]. As a unifying framework, the class of α-fair (also known as "isoelastic") utility functions was proposed for fair allocation in economics in [42]. The main methodology for fair resource allocation in time-varying dynamical systems, akin to the system considered here, is Lyapunov drift analysis. Lyapunov drift has been used as a fundamental design and analysis tool in many problems including the wireless scheduling problem [46, 47], fair resource allocation among competing users [16, 12], stochastic game theory [48]. Based on Lyapunov-drift methods, stochastic dynamic optimization algorithms by using the so-called drift-plus-penalty method were widely used in queueing and networking problems (see [35] and references therein). The existing Lyapunov optimization methods are predominantly opportunistic, which means that the random quantities (such as completion time, reward, system state) arrive prior to the decision-making at each stage, or they assume the knowledge of the first- and second-order statistics of these random quantities. These assumptions are not satisfied in many applications as we discussed in Section 1, therefore the controller must learn the statistics so as to maximize the objective function, such as the total utility. To the best of our knowledge, our paper is the first learning theory approach to the fair resource allocation problem based on Lyapunov drift. Even in the offline optimization setting, the Lyapunov optimization methods are predominantly in discrete-time setting, i.e., each action takes a unit time. The only continuous-time utility maximization approach to fair resource allocation is [19], which assumes the knowledge of first-order statistics. Our work develops an online optimization algorithm based on the Lyapunov optimization methodology proposed in [19] for offline optimization (i.e., when the first-order moments are known), by improving some of the results (e.g., simplified decision rules, finite-time performance bounds), and adapting these for the online learning problem.

The online learning problem under budget constraints has been considered in the bandits with knapsacks (BwK) framework [20]. In this extension of the classical stochastic bandit model, each action consumes a random amount of a resource from a common budget and yields a random reward, where the controller aims to maximize the expected total reward by until a resource is completely depleted. BwK model has been considered under various dynamics [20, 49, 50, 21]. In [23], an interrupt/deadline mechanism is employed to incorporate the continuous-time dynamics into the budget-constrained online learning model. For a detailed discussion of the BwK and its extensions, please refer to [22]. The original BwK models study the reward maximization problem. In [24], the authors consider an online learning setting where the objective is to maximize a concave function subject to convex constraints. In [24], the decision-making process continues for a *fixed* number of stages, and the constraints are not always satisfied unlike our model. Instead, the distance to the constraint set, as well as the regret, is shown to vanish in expectation under the proposed learning algorithms, which require solving linear programs at each stage. Another crucial difference is that the deadline mechanism for improving time-efficiency is not incorporated into the decision in [24]. Our paper deviates from this line of work as it proposes a versatile and comprehensive framework for fairness, and incorporates continuous-time dynamics into the decision-making for time-efficiency under strict time constraints. To solve this problem, we propose a learning algorithm with low computational complexity, and prove its efficiency. The design and analysis methodology we followed in this paper based on Lyapunov optimization can be used in many other problem models.

B Proofs of Proposition 1 and Proposition 2

Proof of Proposition 2. Fix any (group, deadline) decision $(k, t) \in [K] \times \mathbb{T}$, and consider the stationary policy $\pi = \pi(P)$ with an arbitrary probability distribution P. Let the number of (k, t) decisions in $[0, B]$ be defined as

$$N_{\pi}^{(k,t)}(B) = \sum_{n=1}^{N_{\pi(P)}(B)} \mathbb{I}\{\pi_n = (k,t)\}.$$

Since each decision is made independently according to the same probability distribution, the number of tasks between two consecutive tasks for which the decision is (k, t) is iid, which implies that $N_\pi^{(k,t)}(B)$ is a regenerative process [34]. Therefore, we can compute the total reward gathered from tasks for which the decision-pair is (k, t) by using renewal theory. In order to accomplish this, we will compute the mean length of a regenerative cycle for each decision (k, t), and then use the renewal theory for tight bounds.

Without loss of generality, consider a regenerative cycle from the beginning (time 0) to the completion of the first task where the decision-pair is (k, t), thus each regenerative cycle contains exactly one task for which the decision-pair is (k, t). Then, for the random variable $M = \sup\{n \geq 0 : \pi_n(P) \neq (k, t)\}$, the number of tasks in a regenerative cycle is $M + 1 \sim Geo\big(P(k, t)\big)$. This construction implies that $\{M = 0\} = \{\pi_1(P) = (k, t)\}$ and $\{M = m\} = \bigcap_{i=1}^m \{\pi_i(P) \neq (k, t)\} \cap \{\pi_{M+1}(P) = (k, t)\}$ for $m > 1$ under the stationary randomized policy $\pi(P)$. Therefore, the length of the regenerative cycle (i.e., the time interval in which there is exactly one completed task with decision-pair (k, t)) is as follows:

$$Y = \sum_{n=1}^{M} \sum_{(k',t')\neq(k,t)} \mathbb{I}\{\pi_n = (k', t')\}(X_{k',n} \wedge t') + (X_{k,M} \wedge t),$$

where $x \wedge y = \min\{x, y\}$ for any two real numbers x, y. Note that Y is a stopped random walk with non-i.i.d. increments and a controlled stopping time $M + 1$. We will compute the expectation of this quantity first. By iterated expectation, we have the following equality:

$$\mathbb{E}[Y] = \sum_{n_0=0}^{\infty} \mathbb{P}(M = n_0)\mathbb{E}[Y|M = n_0]. \tag{7}$$

Note that for any $n_0 \geq 0$, we have:

$$\mathbb{E}[\mathbb{I}\{\pi_n = (k', t')\}|M = n_0] = \mathbb{P}(\pi_n = (k', t')|\pi_n \neq (k, t)) = \frac{P(k', t')}{1 - P(k, t)},$$

for all $n \leq n_0$. Therefore, we have the following identity:

$$\mathbb{E}[Y|M = n_0] = n_0 \sum_{(k',t')\neq(k,t)} \frac{P(k', t')\mu(k', t')}{1 - P(k, t)} + \mu(k, t), \;\; \forall n_0 \leq 0, \tag{8}$$

where $\mu(k, t) = \mathbb{E}[X_{k,1} \wedge t]$. Thus, we have the following:

$$\begin{aligned} \mathbb{E}[Y] &= \sum_{n_0=0}^{\infty} \mathbb{P}(M = n_0)n_0 \sum_{(k',t')\neq(k,t)} \frac{p(k', t')\mu(k', t')}{1 - p(k, t)} + \mu(k, t), \\ &= \mathbb{E}[M] \sum_{(k',t')\neq(k,t)} \frac{p(k', t')\mu(k', t')}{1 - p(k, t)} + \mu(k, t). \end{aligned}$$

from (7). Since $M + 1 \sim Geo(P(k, t))$, we have $\mathbb{E}[M] = \frac{1}{p(k,t)} - 1$. Substituting this into the above identity, we find the expected length of a regenerative cycle under $\pi(P)$ as follows:

$$\mathbb{E}[Y] = \frac{\sum_{k',t')\neq(k,t)} P(k', t')\mu(k', t')}{P(k, t)}.$$

In summary, a decision-pair (k, t) is chosen once in a cycle of Y time units, and yields a reward $R_{k,n}(t)$ under the stationary randomized policy $\pi(P)$. Having specificed mean length of a regenerative cycle and mean reward, we can now compute the reward rate (i.e., reward per unit time) for a decision-pair (k, t) under $\pi(P)$ as follows:

$$r_k(t) = \frac{\mathbb{E}[R_{k,1}(t)]}{\mathbb{E}[Y]} = \frac{P(k, t)\mathbb{E}[R_{k,1}(t)]}{\sum_{(i,t')\in[K]\times\mathbb{T}} P(i, t')\mathbb{E}[\min\{X_{i,1}, t'\}]}.$$

As an immediate consequence, the reward per unit time for group k under $\pi(P)$ is as follows:

$$\rho_k(P) = \sum_{t\in\mathbb{T}} r_k(t).$$

As a consequence of the elementary renewal theorem [51], the total reward for group k under $\pi(P)$ in $[0, B]$ is $B\rho_k(P) + o(B)$. In order to get tight bounds, we use Lorden's inequality to obtain the following inequalities:

$$B\rho_k(P) \leq \sum_{n=1}^{N_\pi(B)} \sum_{t\in\mathbb{T}} \mathbb{I}\{\pi_n = (k,t)\}R_{k,n}(t) \leq B\rho_k(P) + C(k,t),$$

for a constant $C(k,t) < \infty$ since $Var(X_{k,n} \wedge t) < \infty$ and $Var(R_{k,n}(t)) < \infty$ for all $t \in \mathbb{T}$ [34]. Therefore,

$$\rho_k(P) \leq \overline{r}_k^\pi(B) \leq \rho_k(P) + \frac{C(k,t)}{B}.$$

Since U_k is continuously differentiable and concave, we have the following result:

$$U_k(\rho_k(P)) \leq U_k(\overline{r}_k^\pi(B)) \leq U_k(\rho_k(P)) + U_k'\Big(\rho_k(P)\Big)\frac{C(k,t)}{B},$$

which concludes the proof. □

Proof of Proposition 1. First, we will show an approximation to the optimization problem in (3) based on Jensen's inequality.

Lemma 1. *For any $k \in [K]$, $n \geq 1$ and a causal policy π for choosing $(G_n, T_n, (\gamma_{k,n})_{k\in[K]})$, let*

$$\widetilde{X}_{\pi_n,n} = \min\{X_{G_n,n}, T_n\}, \tag{9}$$

$$Z_{\pi_n,n} = \min\{X_{G_n,n}, T_n\} \sum_{m=1}^{K} U_m(\gamma_{m,n}), \tag{10}$$

$$Y_{\pi_n,m,n} = \min\{X_{G_n,n}, T_n\}\gamma_{m,n} - R_{m,n}(T_n)\mathbb{I}\{G_n = m\},\ \forall m \in [K]. \tag{11}$$

Let U^* be the solution to the following optimization problem:

$$\max_{\pi\in\Pi_A} \quad \lim_{N\to\infty} \frac{\sum_{n=1}^{N} \mathbb{E}[Z_{\pi_n,n}]}{\sum_{n=1}^{N} \mathbb{E}[\widetilde{X}_{\pi_n,n}]} \quad \textit{s.t.} \quad \lim_{N\to\infty} \frac{\sum_{n=1}^{N} \mathbb{E}[Y_{\pi_n,m,n}]}{\sum_{n=1}^{N} \mathbb{E}[\widetilde{X}_{\pi_n,n}]} \leq 0,\ \forall m = 1, 2, \ldots, K. \tag{12}$$

where the maximization is over Π_A, the set of all causal policies. Then, we have the following result:

$$\lim_{B\to\infty} \mathtt{OPT}_{\Pi_A}(B) = U^*.$$

Proof. First, under any policy $\pi \in \Pi_A$, the following holds by the definition of $N^\pi(B)$:

$$\sum_{n=1}^{N^\pi(B)-1} \widetilde{X}_{\pi_n,n} < B \leq \sum_{n=1}^{N^\pi(B)} \widetilde{X}_{\pi_n,n}.$$

Since $\widetilde{X}_{\pi_n,n}$ is bounded for all n, we have the following:

$$\lim_{B\to\infty} \overline{r}_k^\pi(B) = \lim_{B\to\infty} \frac{\mathbb{E}\Big[\sum_{n=1}^{N_\pi(B)} \mathbb{I}\{G_n = k\}R_{k,n}(T_n)\Big]}{\mathbb{E}\Big[\sum_{n=1}^{N_\pi(B)} \widetilde{X}_{\pi_n,n}\Big]} \tag{13}$$

By using the asymptotic equality in (13), continuity of U_k, and a direct application of the extended Jensen's inequality (see Lemma 7.6 in [35]), we have $\lim_{B\to\infty} \mathtt{OPT}_{\Pi_A}(B) = U^*$. This enables us to convert the utility maximization problem into a constrained optimization for time-averages. □

Now, we will prove the following:

$$\max_P \sum_{k\in[K]} U_k\big(\rho_k(P)\big) = U^*.$$

Since U^* is optimal asymptotic total utility over $\Pi_A \supset \Pi_S$, we have the following inequality:

$$\max_P \sum_{k\in[K]} U_k\big(\rho_k(P)\big) \leq U^*.$$

By using (13), a direct application of Lemma 1 in [19] implies that there exists an SRP $\pi(P_0)$ that achieves U^*. Proposition 2 implies that

$$\sum_k U_k(\overline{r}_k^{\pi(P')}(B)) \leq \max_P \sum_k U_k(\rho_k(P)) + O(1/B),$$

for any P' and $B > 0$. Thus, we have:

$$U^* = \lim_{B\to\infty} \sum_k U_k\left(\overline{r}_k^{\pi(P_0)}(B)\right) \leq \max_P \sum_k U_k(\rho_k(P)),$$

which implies $U^* = \max_P \sum_k U_k(\rho_k(P))$. □

C Proof of Proposition 3

Let $\mu(k,t) = \mathbb{E}[\min\{X_{k,1}, t\}]$ and $\theta(k,t) = \mathbb{E}[R_{k,1}(t)]$. For the optimal distribution $P^\star$, let

$$C_k = \sum_{k'\neq k} \sum_{t\in\mathbb{T}} P^\star(k',t)\mu(k',t),$$

$P_k^\star = [P^\star(k,t)]_{t\in\mathbb{T}}$ and $p_k = \sum_{t\in\mathbb{T}} P^\star(k,t)$. Then, since $U_k(x)$ is an increasing function of x, $P_k^\star$ is the solution to the following optimization problem:

$$\max_{P_k} \frac{\sum_t P_k(t)\theta(k,t)}{\sum_t P_k(t)\mu(k,t) + C_k} \quad \text{subject to} \quad P_k(t) \geq 0, \forall t, \quad \sum_t P_k(t) = p_k. \tag{14}$$

Let V^* be the optimum solution of (14), and $V(P_k) = \sum_t P_k(t)\theta(k,t) - V^*\Big(\sum_t P_k(t)\mu(k,t) + C_k\Big)$. Then, the following optimization problem is equivalent to (3):

$$\max_{P_k} V(P_k) \quad \text{subject to} \quad P_k(t) \geq 0, \forall t, \quad \sum_t P_k(t) = p_k, \tag{15}$$

which, in turn, yields $P_k^\star$. For any $t \in \mathbb{T}$, we have $\frac{\partial V}{\partial P_k(t)} = \theta(k,t) - V^*\mu(k,t)$. Let

$$d^* = \max_t \frac{\partial V(P_k)}{\partial P_k(t)}\Big|_{P_k = P_k^\star}.$$

By the optimality of $P_k^\star$, if $P_k^\star(t) > 0$, then we must have $\partial V(P_k^\star)/\partial P_k(t) = d^*$, which further implies that

$$P_k^\star(t) > 0 \Rightarrow \theta(k,t) = d^* + V^*\mu(k,t). \tag{16}$$

Let $t_1 \leq t_2 \leq \ldots \leq t_m$ be the set of deadlines such that $P_k^\star(t_i) > 0$. There exists a $\beta \in [0,1]$ such that the following holds:

$$\sum_t P_k^\star(t)\theta(k,t) = p_k\Big(\beta\theta(k,t_1) + (1-\beta)\theta(k,t_m)\Big).$$

In conjunction with (16), this implies that:

$$\sum_t P_k^\star(t)\mu(k,t) = p_k\Big(\beta\mu(k,t_1) + (1-\beta)\mu(k,t_m)\Big).$$

Hence, we have shown that $P_k^\star$ makes a randomization between at most two deadlines, which simplifies (15) considerably as a function of a single variable $\beta \in [0,1]$. Rewriting (15) in terms of β and taking the derivative with respect to $\beta \in [0,1]$, we observe that the objective function is either monotonically decreasing or increasing with β. Therefore, $P_k^\star$ has only one non-zero element, i.e., the deadline decision is made deterministically for group k.

D Proof of Proposition 4

By Proposition 3, for each group k, there is a unique optimal deadline t_k^*. Let

$$r_k^* = \frac{\mathbb{E}[R_{k,n}(t_k^*)]}{\mathbb{E}[\min\{X_{k,n}, t_k^*\}]},$$

be the reward per processing time for group k under the optimal deadline selection. Then, by Proposition 2, we can express the reward per unit time as follows:

$$\rho_k(P) = r_k^* \widehat{\varphi}_k(P),$$

where

$$\widehat{\varphi}_k(P) = \frac{P(k, t_k^*)\mathbb{E}[\min\{X_{k,n}, t_k^*\}]}{\sum_{j\in[K]} P(j, t_j^*)\mathbb{E}[\min\{X_{j,n}, t_j^*\}]},$$

is the fraction of time allocated to group k under $\pi(P)$. Note that for any P, $\{\widehat{\varphi}_k(P) : k \in [K]\}$ defines a probability distribution in the K-dimensional simplex. Therefore, by Proposition 2, the asymptotically optimal utility is the solution to the following optimization problem:

$$\begin{aligned} \max_{\varphi\in\mathbb{R}_+^K} \sum_{k\in[K]} U_k(r_k^*\widehat{\varphi}_k) \quad \text{s.t.} \quad & \sum_{k\in[K]} \widehat{\varphi}_k = 1, \\ & \widehat{\varphi}_k \geq 0, \ \forall k \in [K]. \end{aligned} \tag{17}$$

The Lagrangian function associated with (17) is as follows:

$$\mathcal{L}(\widehat{\varphi}, \lambda) = \sum_{k\in[K]} U_k(r_k^*\widehat{\varphi}_k) - \lambda\Big(\sum_{k\in[K]} \widehat{\varphi}_k - 1\Big).$$

Since U_k is a monotonically increasing and continuously differentiable function for all k, by solving $\frac{\partial\mathcal{L}}{\partial\widehat{\varphi}_k} = 0$, we obtain $\widehat{\varphi}_k = (U_k')^{-1}(\lambda/r_k^*)$. As U_k is concave for all k, the proof follows by applying KKT conditions.

E Proof of Theorem 1

The proof of Theorem 1 consists of two steps. In the first step, we analyze the performance of the `OLUM` Algorithm for the constrained optimization of time averages for any number of trials N by using a Lyapunov optimization methodology [19, 35, 48]. In the second step, we show that the number of tasks processed in $[0, B]$ is $O(B)$ with high probability to prove the regret result.

The following concentration inequality will be used extensively throughout the proof.

Lemma 2 ([52, 23])**.** *Let X_n and R_n be two sub-Gaussian random processes with means $\mathbb{E}[X] > 0$, $\mathbb{E}[R]$, and parameters σ_X^2 and σ_R^2, respectively. Then, for any $\epsilon \in (0, \mathbb{E}[X])$, we have the following:*

$$\mathbb{P}\Big(\Big|\frac{\sum_{i=1}^n R_i}{\sum_{i=1}^n X_i} - \frac{E[R]}{E[X]}\Big| > \frac{\epsilon(1+r)}{\mu}\Big) \leq 2\big(e^{-n\epsilon^2/\sigma_X^2} + e^{-n\epsilon^2/\sigma_R^2}\big), \tag{18}$$

for any $r > \frac{\mathbb{E}[R]}{\mathbb{E}[X]}$ and $\mu \leq \mathbb{E}[X] - \epsilon$.

Note that any bounded random variable $Z \in [0, a]$ is sub-Gaussian with parameter $\sigma^2 = a^2/4$ [53]. As we are dealing with bounded $\min\{X_{k,n}, t\}$ and $R_{k,n}(t)$, Lemma 2 is an essential result for the proofs in this section.

In the second lemma, we provide an upper bound for the expectation of the dual variables $Q_n = (Q_{1,n}, Q_{2,n}, \ldots, Q_{K,n})$.

Lemma 3. *Consider the dual variables defined in* (6) *under the `OLUM` Algorithm, and without loss of generality assume $Q_{k,0} = 1$ for all k. Then, we have the following bound for any $n \geq 1$:*

$$\mathbb{E}[\sum_{k=1}^K Q_{k,n}] \leq V \sum_{k\in[K]} U_k'\Big(\frac{\min_{k,t}\mathbb{E}[R_{k,n}(t)] - \epsilon}{\max_{k\in[K]}\mathbb{E}[X_{k,n}]}\Big) + O(1/\epsilon), \tag{19}$$

for any $V > 0$ and $\epsilon \in \big(0, \min_{k,t}\mathbb{E}[R_{k,n}(t)]\big)$.

Proof. For any $\epsilon \in \big(0, \min_{k,t} \mathbb{E}[R_{k,n}(t)\big)$, let

$$A = \{q : \sum_k q_k \geq V \sum_{k\in[K]} U_k'\Big(\frac{\min_{k,t} \mathbb{E}[R_{k,n}(t)] - \epsilon}{\max_{k\in[K]} \mathbb{E}[X_{k,n}]}\Big) + \max_t \overline{R}_{max}(t)\}.$$

Then, we have $\mathbb{E}[\sum_k Q_{k,n+1} - \sum_k Q_{k,n}; Q_n \in A\big|Q_n] \leq -\epsilon$. Also, note that $Q_{k,n+1} - Q_{k,n}$ is bounded almost surely, i.e., sub-Gaussian. Thus, Theorem 2.3 in [54] implies the tail bounds for $\sum_k Q_{k,n}$, which implies the result via $\mathbb{E}[X\mathbb{I}\{X > a\}] = a\mathbb{P}(X > a) + \int_a^\infty \mathbb{P}(X > x)dx$. □

Step 1: Recall the equivalent form of the utility maximization problem in Lemma 1. In this step, we will prove the following result under the OLUM Algorithm:

$$\frac{\mathbb{E}[\sum_{n=1}^N Z_{\pi_n,n}]}{\mathbb{E}[\sum_{n=1}^N \widetilde{X}_{\pi_n,n}]} \geq U^* - O\Big(\sqrt{\frac{\log(N)}{N}} + \frac{1}{V}\Big),$$

$$\frac{\mathbb{E}[\sum_{n=1}^N Y_{\pi_n,m,n}]}{\mathbb{E}[\sum_{n=1}^N \widetilde{X}_{\pi_n,n}]} \leq O(V/N),\ m = 1, 2, \dots, K.$$

for any N. This will be done by showing that the OLUM Algorithm achieves ϵ-optimal Lyapunov drift with high probability for each decision, thus achieves optimality fast as a result of the Lyapunov drift methodology. For details on Lyapunov optimization, refer to [35].

For any group $k \in [K]$, let

$$\begin{aligned} X_{k,n}^* &= \min\{X_{G_n,n}, t_k^*\}, \\ Z_{k,n}^* &= \min\{X_{k,n}, t_k^*\} \sum_{m=1}^K U_m(\gamma_{m,n}), \\ Y_{k,m,n}^* &= \min\{X_{k,n}, t_k^*\}\gamma_{m,n} - R_{m,n}(t_m^*)\mathbb{I}\{k = m\},\ \forall m \in [K]. \end{aligned}$$

Note that these are the random variables in Lemma 1 under the optimal deadline t_k^* for each group k.

The proof relies on a novel online learning approach based on drift-based optimization techniques. In this methodology, the dual variables Q_n as defined in (6) summarize how much the constraint is violated in the past. At stage n, given the vector of dual variables Q_n, we have the drift-plus-penalty ratio (DPPR), which is defined as follows:

$$\Psi_n(k, Q_n) = -V\frac{\mathbb{E}[Z_{k,n}^*]}{\mathbb{E}[\min\{X_{k,n}, t_k^*\}]} + \sum_m Q_{m,n}\frac{\mathbb{E}[Y_{k,m,n}^*]}{\mathbb{E}[\min\{X_{k,n}, t_k^*\}]}. \tag{20}$$

The optimal algorithm therefore, aims to minimize the DPPR to achieve optimality. Let the terms in DPPR related to the auxiliary variables $\gamma_{m,n}$ be denoted as:

$$\psi_n(\gamma_n, Q_n) = \sum_{m=1}^K \big(-VU_m(\gamma_{m,n}) + Q_{m,n}\gamma_{m,n}\big). \tag{21}$$

Therefore, the DPPR can be written as follows:

$$\Psi_n(k, Q_n) = \psi_n(\gamma_n, Q_n) - Q_{k,n}\frac{\mathbb{E}[R_{k,n}(t_k^*)]}{\mathbb{E}[\min\{X_{k,n}, t_k^*\}]}. \tag{22}$$

The classical drift-based stochastic optimization techniques either assume the knowledge of the first-order moments in $\Psi_n(k, G_n)$, or they observe the outcomes for the completion of task n prior to the decision. However, in online learning, since we have no prior knowledge of the mean values $\mathbb{E}[R_{m,n}(t_m^*)]$ and $\mathbb{E}[\min\{X_{k,n}, t_k^*\}]$, we define the empirical reward-per-processing-time as follows:

$$\widehat{r}_{k,n}(t) = \frac{\sum_{i=1}^{n-\tau} R_{k,i}(t)}{\sum_{i=1}^{n-\tau} \min\{X_{k,i}, t\}}. \tag{23}$$

where $n - \tau$ is the number of samples available. Similarly, let

$$r_k(t) = \frac{\mathbb{E}[R_{k,i}(t)]}{\mathbb{E}[\min\{X_{k,i}, t\}]}. \tag{24}$$

The deadline is chosen so as to maximize the reward per processing time:

$$\widehat{r}_{k,n} = \max_{t \in \mathbb{T}} \widehat{r}_{k,n}(t).$$

Let $\delta_k(t) = \max_{t'} r_k(t') - r_k(t)$ and $\delta(t) = \min_{(k,t):\delta_k(t)>0} \delta_k(t)$. By using Lemma 2, it can be shown that $T_n = t^*_{G_n}$ with probability at least $1 - e^{-n\Omega\left(\delta^2(t)\right)}$, i.e., the optimal deadline for the chosen group G_n is selected with high probability. With this deadline-selection policy, the empirical drift-plus-penalty ratio (e-DPPR) is defined as follows:

$$\widehat{\Psi}_n(k, Q_n) = \psi_n(\gamma_n, Q_n) - Q_{k,n}\widehat{r}_{k,n}. \tag{25}$$

The OLUM Algorithm as defined in Definition 4 is based on minimizing the e-DPPR in (25). The auxiliary variables in the OLUM Algorithm is chosen to maximize $\psi_n(\gamma_n, Q_n)$ over γ_n, and the group decision is independent of the choice of the auxiliary variables given Q_n.

The following proposition quantifies the approximation error for using the e-DPPR in the decision-making as a surrogate for the DPPR in the optimization.

Proposition 6. *For any given $\epsilon \in (0, \mu_*)$, we have the following inequality for the DPPR under the* OLUM *Algorithm:*

$$\Psi_n(G_n, Q_n) \leq \min_{k \in [K]} \Psi_n(k, Q_n) + \frac{2\epsilon(1 + r^*)}{\mu_* - \epsilon} \sum_k Q_{k,n} + h(Q_n)O(n), \tag{26}$$

where $\mathbb{E}[h(Q_n)] = c_1 e^{-c_2 n \epsilon^2}$ for some constants $c_1, c_2 > 0$ and

$$r^* = \max_{(k,t)} \frac{\mathbb{E}[R_{k,n}(t)]}{\mathbb{E}[\min\{X_{k,n} t\}]}.$$

The proof of Proposition 6 relies on the concentration result presented in Lemma 2 and a PAC-type bound: let k be a group such that $\Psi_n(k, Q_n) > \min_j \Psi_n(j, Q_n) + \delta$ for any $\delta > 0$ given Q_n. Then,

$$\begin{aligned}\mathbb{P}\big(G_n = k\big|Q_n\big) \leq \mathbb{P}\big(\big|\widehat{\Psi}_n(k, Q_n) - \Psi_n(k, Q_n)\big| > \delta/2\big|Q_n\big) \\ + \mathbb{P}\big(\big|\widehat{\Psi}_n(k_n, Q_n) - \Psi_n(k_n, Q_n)\big| > \delta/2\big|Q_n\big),\end{aligned}$$

where $k_n = \arg\min_j \Psi_n(j, Q_n)$. Then, a straightforward application of Lemma 2 and union bound (over suboptimal groups) with $\delta = \epsilon \cdot O(\sum_k Q_{k,n})$ for $\epsilon > 0$ yield the result.

We have the following lemma, which will be key in the analysis of the learning algorithm.

Lemma 4 ([35])**.** *Let $L(q) = \frac{1}{2}\sum_{m=1}^K q_m^2$ be the quadratic Lyapunov function, and*

$$\Delta(Q_n) = \mathbb{E}[L(Q_{n+1}) - L(Q_n)|Q_n],$$

be the Lyapunov drift. Then, we have the following bound on the Lyapunov drift for the problem (12)*:*

$$\Delta(Q_n) \leq D + \mathbb{E}[\sum_{k \in [K]} Q_{k,n} Y_{G_n,k,n}|Q_n], \tag{27}$$

for some constant $D > 0$ under the OLUM *Algorithm.*

Following the methodology in [19], from Prop. 6 and Lemma 4 with $\epsilon = \epsilon_n = \frac{2(1+r^*)}{\mu_*}\sqrt{\frac{\beta \log(n)}{n}}$ for $\beta > 2$, we have the following result:

$$\begin{aligned}\Delta(Q_n) - V\mathbb{E}[\min\{X_{G_n,n}, T_n\} \sum_k U_k(\gamma_{k,n})|Q_n] \leq D \\ + \mathbb{E}[\min\{X_{G_n,n}, T_n\}|Q_n]\Big(-VU^* + \epsilon_n \sum_k Q_{k,n} + \mathbb{E}[h(Q_n)|Q_n]O(K \cdot n)\Big).\end{aligned} \tag{28}$$

where $D > 0$ is a constant, and the RHS holds since there exists an optimal stationary randomized policy for (12) which satisfies:

$$\min_k \Psi(k, Q_n) \leq \Psi(\tilde{G}_n, Q_n) = -VU^*.$$

Taking the expectation in (28), we have:

$$\mathbb{E}[L(Q_{n+1}) - L(Q_n)] - V\mathbb{E}[\min\{X_{G_n,n}, T_n\}\sum_k U_k(\gamma_{k,n})] \leq B - VU^*\mathbb{E}[\min\{X_{G_n,n}, T_n\}] \\ + \epsilon_n \max_{k\in[K]} \mathbb{E}[X_{k,n}] \sum_{k\in[K]} \mathbb{E}[Q_{k,n}] + O(K)n^{1-\beta}, \quad (29)$$

Summing the above over $n = 0, 1, \dots, N-1$, dividing by N, and rearranging terms, we have the following inequality:

$$\frac{\mathbb{E}[\sum_{n=1}^N Z_{\pi_n,n}]}{\mathbb{E}[\sum_{n=1}^N \widetilde{X}_{\pi_n,n}]} \geq U^* - \frac{2(1+r^*)}{\mu_*} O\Big(\sqrt{\frac{\beta\log(N)}{N}}\Big) + \frac{D/\mu_* + O(N^{\beta-2})}{V} + \frac{\mathbb{E}[L(Q_0)]}{V\mu_* N}. \quad (30)$$

The second question we had was how much the constraint in (12) is violated. From the update of the dual variables (6), we have the following:

$$Q_{k,n+1} \geq Q_{k,n} + Y_{\pi_n,k,n}, \quad (31)$$

Summing the above over all $n = 0, 1, \dots, N-1$, we have:

$$Q_{k,N} \geq Q_{k,0} + \sum_{n=1}^N Y_{G_n,k,n}.$$

Thus, we have:

$$\frac{\mathbb{E}[Q_{k,N}]}{N\mu_*} \geq \frac{\mathbb{E}[\sum_{n=1}^N Y_{\pi_n,k,n}]}{\mathbb{E}[\sum_{n=1}^N \min\{X_{G_n,n}, T_n\}]}. \quad (32)$$

By Lemma 3, the following inequality holds:

$$\frac{\mathbb{E}[Q_{k,N}]}{N} \leq O\Big(\frac{V}{N}\Big).$$

Hence, by choosing $V = \Theta(\sqrt{N/\log(N)})$, we show that the objective is achieved with $O(\sqrt{\log(N)/N})$ gap, and the constraint is satisfied at a rate $O(1/\sqrt{N\log(N)})$.

Step 2. In this step, we will show that the decision-making process continues for $N^\pi(B) = \Theta(B)$ stages with high probability, which will conclude the proof.

For any $B > 0$, let $n_0(B) = \lceil 2B/\mu_{min} \rceil$. Then, under any causal policy π, we have the following bound:

$$\begin{aligned}\mathtt{REG}^\pi_{\Pi_S}(B) &= U^* - \frac{\mathbb{E}[\sum_{n=1}^{N^\pi(B)} Z_{\pi_n,n}]}{B}, \\ &\leq U^* - \frac{\mathbb{E}[\sum_{n=1}^{N^\pi(B)} Z_{\pi_n,n}]}{\mathbb{E}[\sum_{n=1}^{N^\pi(B)} \widetilde{X}_{\pi_n,n}]}, && (33) \\ &\leq \frac{\mu^* \cdot n_0(B)}{B}\Big(U^* - \frac{\mathbb{E}[\sum_{n=1}^{n_0(B)} Z_{\pi_n,n}]}{\mathbb{E}[\sum_{n=1}^{n_0(B)} \widetilde{X}_{\pi_n,n}]} + o(1)\Big), && (34)\end{aligned}$$

where $U^* = \mathtt{OPT}_{\Pi_S}(B) + O(1/B)$ is the optimal utility in Lemma 1, $\mu^* = \max_k \mathbb{E}[X_{k,n}]$, and (33) holds since $\sum_{n=1}^{N^\pi(B)} X_{G_n,n} \geq B$ by definition. In order to prove (34), first note that

$$\begin{aligned}\mathbb{E}\Big[\sum_{n=1}^{N^\pi(B)} \big(U^*\widetilde{X}_{\pi_n,n} - Z_{\pi_n,n}\big)\Big] &= \mathbb{E}\Big[\sum_{n=1}^{\infty} \big(U^*\widetilde{X}_{\pi_n,n} - Z_{\pi_n,n}\big)\mathbb{I}\{N^\pi(B) > n\}\Big], \\ &\leq \mathbb{E}\Big[\sum_{n=1}^{n_0(B)} \big(U^*\widetilde{X}_{\pi_n,n} - Z_{\pi_n,n}\big)\Big] + U^* \sum_{n > n_0(B)} \mathbb{P}(N^\pi(B) > n)\end{aligned} \quad (35)$$

Since $\{N^{\pi}(B) > n\} \subset \{\sum_{i=1}^{n} \widetilde{X}_{\pi_i,i} < B\}$ by definition and $\mathbb{E}[\widetilde{X}_{\pi_i,i}|\mathcal{H}_i] \geq \mu_* > 0$ for all i, we have:

$$\mathbb{P}(N^{\pi}(B) > n) = \mathbb{P}(\sum_{i=1}^{n} \widetilde{X}_{\pi_i,i} < B) \leq e^{-n\Omega(1)},$$

for all $n > n_0(B)$ by Azuma-Hoeffding inequality [53], which implies that $n_0(B)$ is a high-probability upper bound for $N^{\pi}(B)$ under any causal policy π. In other words, the decision-making process continues for at most $n_0(B)$ turns with high probability since each action depletes a positive amount from the time budget B. Consequently, we have

$$\sum_{n > n_0(B)} \mathbb{P}(N^{\pi}(B) > n) \leq e^{-\Omega(B)} = o(1).$$

Using this result and rearranging the terms in (35), we obtain the inequality in (34). Furthermore, the constraints are satisfied at rate $O(V/B)$ for all groups. Therefore, by using the result of Step 1 with $N = n_0(B)$ and noting that $n_0(B)/B = \Theta(1)$, we conclude that $\texttt{REG}_{\Pi_S}^{\pi}(B) = O(\sqrt{\log(B)/B})$.