

Learning to Schedule Online Tasks with Bandit Feedback

Yongxin Xu
ShanghaiTech University
Shanghai, China
xuyx2022@shanghaitech.edu.cn

Shangshang Wang
ShanghaiTech University
Shanghai, China
wangshsh2@shanghaitech.edu.cn

Hengquan Guo
ShanghaiTech University
Shanghai, China
guohq@shanghaitech.edu.cn

Xin Liu
ShanghaiTech University
Shanghai, China
liuxin7@shanghaitech.edu.cn

Ziyu Shao
ShanghaiTech University
Shanghai, China
shaozy@shanghaitech.edu.cn

ABSTRACT

Online task scheduling serves an integral role for task-intensive applications in cloud computing and crowdsourcing. Optimal scheduling can enhance system performance, typically measured by the reward-to-cost ratio, under some task arrival distribution. On one hand, both reward and cost are dependent on task context (e.g., evaluation metric) and remain black-box in practice. These render reward and cost hard to model thus unknown before decision making. On the other hand, task arrival behaviors remain sensitive to factors like unpredictable system fluctuation whereby a prior estimation or the conventional assumption of arrival distribution (e.g., Poisson) may fail. This implies another practical yet often neglected challenge, i.e., uncertain task arrival distribution. Towards effective scheduling under a stationary environment with various uncertainties, we propose a double-optimistic learning based Robbins-Monro (DOL-RM) algorithm. Specifically, DOL-RM integrates a learning module that incorporates optimistic estimation for reward-to-cost ratio and a decision module that utilizes the Robbins-Monro method to implicitly learn task arrival distribution while making scheduling decisions. Theoretically, DOL-RM achieves a sub-linear regret of $O(T^{3/4})$, which is the first result for online task scheduling under uncertain task arrival distribution and unknown reward and cost. Our numerical results in a synthetic experiment and a real-world application demonstrate the effectiveness of DOL-RM in achieving the best cumulative reward-to-cost ratio compared with other state-of-the-art baselines.

KEYWORDS

Online scheduling, Double-Optimistic learning, Robbins-Monro method

ACM Reference Format:

Yongxin Xu, Shangshang Wang, Hengquan Guo, Xin Liu, and Ziyu Shao. 2024. Learning to Schedule Online Tasks with Bandit Feedback. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024)*, Auckland, New Zealand, May 6 – 10, 2024, IFAAMAS, 9 pages.



This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

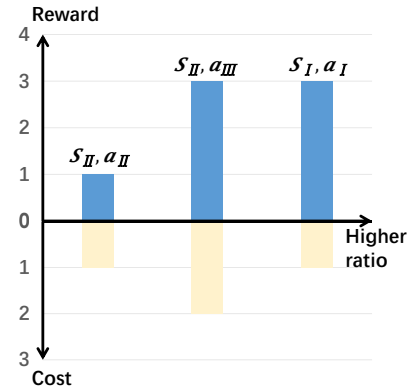


Figure 1: Rewards and Costs for (S, a) in a Toy System.

Table 1: Reward-to-Cost Ratio under Distribution $\{0.8, 0.2\}$.

Algorithm	Policy	Ratio
The greedy algorithm	$(S_I, a_I), (S_{II}, a_{III})$	2.5
The reverse algorithm	$(S_I, a_I), (S_{II}, a_{II})$	2.6

1 INTRODUCTION

Online task scheduling refers to the process that schedules incoming tasks to available resources in real-time. It plays a central role in boosting operational efficiency for cloud computing [1, 3, 16], crowdsourcing [2, 9, 22] and multiprocessing [11, 17, 21] systems, etc. When a task is scheduled, it occupies resources for processing, e.g., computing hardware or labor costs. Upon task completion, we receive rewards, e.g., a high-accuracy machine learning model or high-quality data. The typical goal of online task scheduling is to optimize the *reward-to-cost ratio*, which represents the system efficiency. For instance, a cloud computing platform bills users to process their tasks using virtualized processing units, aiming to maximize the average revenue-to-hardware cost ratio. A crowdsourcing platform compensates appropriate workers for processing data labeling tasks, aiming to optimize the quality-to-cost ratio.

There exist two major challenges to establish an effective online task scheduling policy. The first one is the black-box nature of rewards and costs. Take the machine learning model training as an example, rewards could refer to the model training accuracy, area under the curve (AUC) and $f1$ score while the costs could refer to the power consumption [20, 32]. The scheduler needs to choose

an appropriate training time (decision) to maximize the accuracy (reward) per power consumption (cost). However, the knowledge of accuracy and power consumption w.r.t. training time is unknown a priori and thus information is only revealed for the chosen decision (i.e., bandit feedback) when the task is completed.

The second challenge is the uncertainty of task arrival, where there exist multiple types of tasks in the system and the scheduler has no prior information on their arrival distribution. One intuitive approach to addressing this challenge is always choosing the greedy decision with the maximal estimated ratio (suppose the knowledge of rewards and costs is known), regardless of the task type. This idea seems to work because the greedy decision for every incoming tasks would maximize the average reward-to-cost across all tasks. However, we show the greedy decision could be an arbitrarily sub-optimal solution by slightly twisting the task arrival distribution. Consider a toy task scheduling system with arriving tasks of two types in Figure 1, denoted as task S_I and S_{II} . For task S_I , there is only one available decision a_I . For task S_{II} , two decisions exist, i.e., the first decision a_{II} involves a low reward and a low cost, while the second one a_{III} is characterized by a higher reward and a higher cost which has a higher reward-to-cost ratio than decision a_{II} . We consider two algorithms, i.e., the greedy (a_I for task S_I and a_{III} for task S_{II}) and the reverse (a_I for task S_I and a_{II} for task S_{II}) algorithm. One may argue that the greedy algorithm is more favorable since it achieves a higher ratio for task S_{II} . However, when the task arrival distribution is $\{0.8, 0.2\}$ in Table 1, this intuitive algorithm achieves an expected ratio of $(0.8 \times 3 + 0.2 \times 3)/(0.8 \times 1 + 0.2 \times 2) = 2.5$, which is worse than its reverse algorithm with that of $(0.8 \times 3 + 0.2 \times 1)/(0.8 \times 1 + 0.2 \times 1) = 2.6$. One may then suggest to adopt the reverse algorithm instead. However, we can always construct another distribution (e.g., $\{0.2, 0.8\}$) to fail the reverse algorithm such that it turns out to be worse. From this example, we emphasize the critical role of task arrival distribution in establishing effective online task scheduling algorithms.

To address the two challenges above, we propose a double-optimistic learning approach to estimate the reward and cost with bandit feedback. Intuitively, the learning approach establishes the optimistic and pessimistic estimations for rewards and costs, respectively, yielding an overall optimistic estimation for the reward-to-cost ratio. This confirms the principle, the optimism in the face of uncertainty, which is exemplified by the confidence bound based algorithms [24]. Furthermore, instead of adopting naive estimation for the task arrival distribution, we utilize the Robbins-Monro method to implicitly learn the task arrival distribution while making decisions. Intuitively, this method transforms the problem of reward-to-cost ratio maximization into a fixed point problem, which can be efficiently solved by carefully using stochastic samples and yields a fast convergence.

In summary, we propose and analyze a novel optimization framework for online task scheduling, where the objective is to optimize the cumulative reward-to-cost ratio under a stationary environment with uncertain task arrival distribution. We integrate double-optimistic learning and the Robbins-Monro method for effective and efficient online scheduling. Our main contributions are summarized as follows:

- **Model for Online Task Scheduling.** We propose a general framework for online task scheduling without any prior knowledge

of reward, cost and task arrival distribution. This framework encapsulates a variety of practical instances whose goal is to optimize the reward-to-cost ratio, thereby striving to achieve high-return and cost-effective outcomes in real-world scheduling systems.

- **Algorithm Design.** We propose a novel algorithm called DOL-RM which incorporates double-optimistic learning for unknown rewards and costs and a modified Robbins-Monro method to implicitly learn the uncertain task arrival distribution. This integrated design enables the balance between rewards and costs such that the cumulative reward-to-cost ratio is maximized.

- **Theoretical Analysis.** We prove that DOL-RM achieves a sub-linear regret at order $O(T^{3/4})$ against the optimal scheduling policy in hindsight (Theorem 4.1). To prove the main result, we decompose the regret into two individual errors w.r.t. double-optimistic learning and Robbins-Monro method and then leverage the Lyapunov drift technique and carefully control the cumulative errors over the entire learning process.

- **Applications.** We test DOL-RM and compare it with state-of-the-art baselines via both a synthetic simulation and a real-world experiment of machine learning task scheduling. These results demonstrate that DOL-RM can achieve the best cumulative reward-to-cost ratio without any prior knowledge of reward, costs and task arrival distribution.

1.1 Related Work

Online Task Scheduling. Online task scheduling has been studied extensively in the literature. We focus on presenting the works most related to ours. In [15], the authors utilize a UCB variant to address the exploration-exploitation dilemma in online task scheduling, however, the work does not take the cost into the consideration. In [26], a Robbins-Monro based approach is proposed for task scheduling in edge computing. However, its primary goal is to preserve data freshness, measured by Age of Information (AoI), differs from our setting of maximizing a generic reward-to-cost ratio. In [5], online bandit feedback is leveraged to schedule tasks in a renewal system, which is distinct from ours because it allows the scheduler to interrupt a task in service but the task in our model cannot be stopped once being scheduled. The closely related work is [31], which studies a similar setting with this paper except assuming the decision set and the information of reward and cost are completely known before decision making. In contrast, we focus on a more practical scenario without assuming any knowledge of the rewards and costs. Moreover, our algorithm and proof techniques are also different with [31] as we need to handle the additional uncertainties of the rewards and costs.

Cost-Aware Bandits: The key feature in our online task scheduling problem is that costs are concomitant with rewards upon decision-making. This feature is also captured in another decision-making model, i.e., the cost-aware bandit model, where an agent incurs a reward and a cost simultaneously by pulling an arm. The cost-aware bandits can be specialized into two major models, including budget-constrained bandits and the cost-aware cascading bandits. The budget-constrained bandits have been widely explored in [6, 8, 10, 25, 40, 41, 43–45], where pulling an arm incurs an additional cost and there exists a hard stopping point once the cumulative cost exceeds a given budget. Our setting has two major differences with

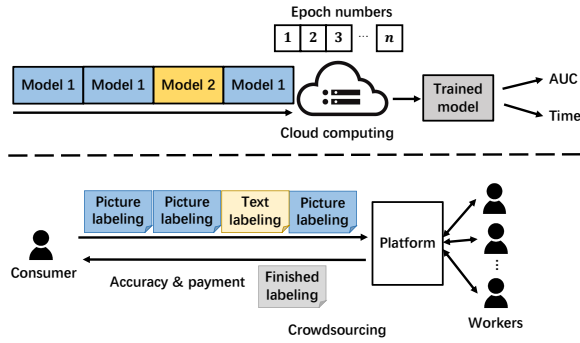


Figure 2: Online Task Scheduling in Cloud Computing (above) and Crowdsourcing (bottom).

the budget-constrained bandits. The first one is we do not have or assume any explicit budget limit constraints; and the second one is we consider a “continual” system, where it would not stop in the middle until all tasks are completed. Therefore, the previous algorithms and analysis in budget-constrained bandits cannot be applied in this paper. Besides, cost-aware cascading bandits are also related and have been studied in [7, 13, 14, 37, 42]. Its goal is to maximize the reward-to-cost gap, however, such a maximal gap does not necessarily yield the optimal ratio as in our paper because we have multiple types of tasks in the system and need to take the task arrival distribution into consideration.

Compared to the previous works, this paper takes a bold step and investigates online task scheduling without any knowledge of rewards, costs, and task arrival distribution. Accordingly, we propose DOL-RM, an effective algorithm for optimizing the reward-to-cost ratio in online task scheduling. Furthermore, DOL-RM establishes a near-optimal regret performance for this challenging setting and has been validated to achieve the best empirical performance via a synthetic simulation and a real-world experiment.

2 SYSTEM MODEL

We study a typical online task scheduling system where a central controller processes an incoming sequence of heterogeneous tasks in a back-to-back manner (i.e., the controller observes and processes the next task immediately upon the completion of its current task). We assume S types of incoming tasks in the system, denoted by the set $\mathcal{S} \triangleq \{1, 2, \dots, S\}$ and every task is randomly drawn from a stationary probability distribution $\mathbb{P} : \mathcal{S} \rightarrow \mathbb{R}^+$. For the t th task, the controller observes its type $S_t \in \mathcal{S}$ and chooses a decision $a_t \in \mathcal{A}(S_t)$, where $\mathcal{A}(S_t)$ is the corresponding available decision set of task type S_t . When the t th task is completed, the controller observes the feedback of reward R_{S_t, a_t} and cost C_{S_t, a_t} , which are randomly generated from with the expected values $r_{S_t, a_t} \triangleq \mathbb{E}[R_{S_t, a_t}]$ and $c_{S_t, a_t} \triangleq C_{S_t, a_t}$. Note we only observe the feedback with respect to the selected decision a_t , thus coined as “bandit feedback”, a term borrowed from bandit learning [24]. Then, the controller continues to work on the next $(t + 1)$ th task until T tasks are completed. In the paper, we assume the rewards $\{R_{S_t, a_t}\}_t$ and costs $\{C_{S_t, a_t}\}_t$ are independent if (S_t, a_t) is the same.

Reward-to-Cost Ratio Maximization: The controller aims to maximize the cumulative *reward-to-cost ratio* over a sequence of T tasks as follows:

$$\max_{\{a_t\}_t} \frac{\sum_{t=1}^T \mathbb{E}[R_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[C_{S_t, a_t}]} \tag{1}$$

Expectation in Problem (1) is taken over the randomness w.r.t. reward, cost and task arrival distribution $\mathbb{P}(\cdot)$. We emphasize that we do not assume any prior knowledge on such distribution and instead learn it implicitly in an online manner (please refer to Section 3).

When the task arrival distribution and the expected reward and cost are known, we can compute the optimal policy by solving Problem (1) with offline optimization techniques [12]. However, such knowledge is practically unavailable or infeasible to the controller and we have to learn it by exploring the decision space in an online manner. Moreover, as discussed in the introduction, the greedy algorithm that simply maximizes $R_{S_t, a_t} / C_{S_t, a_t}$ for each individual task can achieve arbitrary sub-optimal performance because it ignores the task arrival distribution. Therefore, we propose an online learning based algorithm to address challenges on the uncertainties of rewards, costs and task arrivals.

Before presenting our algorithm, we emphasize that our modeling is general to capture many real-world applications, which is illustrated with two following examples as shown in Figure 2:

- **Machine Learning (ML) model training on cloud servers:** The cloud platforms (e.g., AWS, Microsoft Azure or Google Cloud) constantly process ML model training tasks submitted by users. The types of ML training tasks are unknown a priori and only revealed at the server when they are executed. For each upcoming task, the platform decides a specific training time from a range of options, which corresponds to the cost; when the training process is finished, the test accuracy is returned as the reward. However, the relationship of accuracy v.s. training time is uncertain. The goal of the platform is to schedule the training time for each individual task such that the average accuracy per time unit (i.e., accuracy-to-time ratio) is maximized.
- **Tasks assignment in Crowdsourcing:** The crowdsourcing platforms (e.g., Amazon Turk or Task Rabbit) received a sequence of data labeling tasks submitted by consumers. The platform is unaware of the type of tasks until they arrive. For each task, the platform assigns it to a suitable worker from a set of candidates with various task proficiency and expertise. The quality of the labeled data from workers is considered as the reward, while his/her payment corresponds to the cost. The goal of the platform is to assign the worker for each individual task such that the average quality per dollar (i.e., quality-to-payment ratio) is maximized.

Next, we present our online learning and decision algorithm to solve the reward-to-cost ratio maximization in Problem (1).

3 ALGORITHM DESIGN

In this section, we propose a double-optimistic learning based Robbins-Monro (DOL-RM) algorithm to address the challenges induced by unknown reward, cost and task arrival distributions. Specifically, DOL-RM includes two algorithmic modules, i.e., the *learning module* leverages double-optimistic learning for unknown

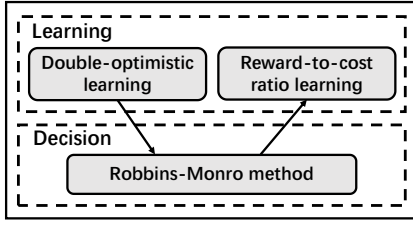


Figure 3: Illustration of DOL-RM Algorithm.

rewards and costs, and the *decision module* utilizes the Robbins-Monro method to make decisions via implicitly learning the task arrival distribution. As shown in Figure 3, upon each task arrives, the learning module first estimates the reward and cost via a double-optimistic learning approach. Based on the estimation, the decision module makes a proper decision through the Robbins-Monro method, which in turn provides an iteratively approaching estimation for the reward-to-cost ratio even without explicitly learning the task arrival distribution. Next, we introduce these two major modules in detail and explain the intuition behind the algorithm. The completed DOL-RM algorithm is depicted in **Algorithm 1**.

Double-Optimistic Learning: Recall that for the t th task with the type S_t , the expected reward $r_{S_t,a}$ and cost $c_{S_t,a}$ for each decision $a \in \mathcal{A}(S_t)$ are unknown. We consider them as multi-armed bandit problems, where each decision is regarded as an arm associated with a pair of reward and cost. Accordingly, we leverage the canonical algorithm design principle in bandit learning [24], optimism in the face of uncertainty, to estimate the reward and cost as follows:

$$\hat{r}_{S_t,a} \triangleq \min \left\{ r_{\max}, \bar{R}_{S_t,a} + \sqrt{\frac{\log T}{N_{S_t,a}(t-1)}} \right\}, \quad (2)$$

$$\check{c}_{S_t,a} \triangleq \max \left\{ c_{\min}, \bar{C}_{S_t,a} - \sqrt{\frac{\log T}{N_{S_t,a}(t-1)}} \right\}, \quad (3)$$

where $\bar{R}_{S_t,a}$ and $\bar{C}_{S_t,a}$ denote the empirical means of reward and cost; $N_{S_t,a}(t-1)$ denotes times the arm (decision) a has been chosen for task of type S_t until round $(t-1)$; r_{\max} and c_{\min} are the maximum reward and minimum cost, respectively. Note that $\hat{r}_{S_t,a}$ and $\check{c}_{S_t,a}$ denote the truncated Upper Confidence Bound (UCB) and Lower Confidence Bound (LCB) for reward and cost estimation, respectively. Recall in Problem (1), $\hat{r}_{S_t,a}$ and $1/\check{c}_{S_t,a}$ serve as the optimistic estimators for $\mathbb{E}[R_{S_t,a}]$ and $1/\mathbb{E}[C_{S_t,a}]$, yielding the name of “Double-Optimistic Learning”. Intuitively, this learning policy encourages exploration among arms (decision space) to learn the optimal reward-to-cost ratio effectively.

Suppose we make decision a_t for the task S_t and observe the reward R_{S_t,a_t} and cost C_{S_t,a_t} feedback when the task is completed, we conduct the following updates:

$$N_{S_t,a_t}(t) = N_{S_t,a_t}(t-1) + 1, \quad (4)$$

$$\bar{R}_{S_t,a_t}(t) = \frac{\bar{R}_{S_t,a_t}(t-1)N_{S_t,a_t}(t-1) + R_{S_t,a_t}}{N_{S_t,a_t}(t)}, \quad (5)$$

$$\bar{C}_{S_t,a_t}(t) = \frac{\bar{C}_{S_t,a_t}(t-1)N_{S_t,a_t}(t-1) + C_{S_t,a_t}}{N_{S_t,a_t}(t)}. \quad (6)$$

Algorithm 1 DOL-RM

Input: Number of tasks T , minimum cost c_{\min} , θ_{\max} and θ_{\min} .

Initialization: $\theta_1 \leftarrow \theta_{\min}$, $\eta \leftarrow \frac{1}{c_{\min}\sqrt{T}}$.

for $t = 1, 2, \dots, T$ **do**

•**Observation:** Task type S_t and available decision set $\mathcal{A}(S_t)$.

•**Double-Optimistic Learning:** Calculate reward and cost estimates $\hat{r}_{S_t,a}$ and $\check{c}_{S_t,a}$, $\forall a \in \mathcal{A}(S_t)$ according to (2) and (3), respectively.

•**Robbins-Monro Based Decision:** Choose the greedy decision for task scheduling:

$$a_t \leftarrow \arg \max_{a \in \mathcal{A}(S_t)} \hat{r}_{S_t,a} - \theta_t \check{c}_{S_t,a} \quad (7)$$

•**Bandit Feedback:** Obtain reward R_{S_t,a_t} and cost C_{S_t,a_t} .

•**Reward-to-Cost Ratio Learning:**

$$\theta_{t+1} \leftarrow \left[\theta_t + \eta (\hat{r}_{S_t,a_t} - \theta_t \check{c}_{S_t,a_t}) \right]_{\theta_{\min}}^{\theta_{\max}}. \quad (8)$$

Update the empirical means of rewards and costs with (4)–(6).

end for

Robbins-Monro Based Decision: Let θ^* be the optimal objective (i.e., reward-to-cost ratio) in Problem (1). It is notable to observe that Problem (1) is equivalent to finding an optimal sequence of actions such that:

$$\max_{\{a_t\}_t} \mathbb{E} \left[\sum_{t=1}^T R_{S_t,a_t} - \theta^* \sum_{t=1}^T C_{S_t,a_t} \right] = 0, \quad (9)$$

This is essentially a fixed point problem and can be solved using the Robbins-Monro method with stochastic samples [30, 34]. Specifically, assuming the optimal ratio θ^* is known and using $\hat{r}_{S_t,a}$ and $\check{c}_{S_t,a}$ as the proxy of R_{S_t,a_t} and C_{S_t,a_t} , we can follow a greedy decision for task $S_t \in \mathcal{S}$ according to (9):

$$\arg \max_{a \in \mathcal{A}(S_t)} \hat{r}_{S_t,a} - \theta^* \check{c}_{S_t,a}$$

However, this decision is non-causal and infeasible due to the lack of knowledge on the best ratio θ^* . To learn θ^* , we modify the Robbins-Monro iteration method by plugging the $\hat{r}_{S_t,a}$ and $\check{c}_{S_t,a}$:

$$\theta_{t+1} = \left[\theta_t + \eta (\hat{r}_{S_t,a_t} - \theta_t \check{c}_{S_t,a_t}) \right]_{\theta_{\min}}^{\theta_{\max}}$$

where η is the learning rate and $[\cdot]_{\theta_{\min}}^{\theta_{\max}}$ denotes the projection of real number onto the interval $[\theta_{\min}, \theta_{\max}]$ with $\theta_{\min} \triangleq r_{\min}/c_{\max}$ and $\theta_{\max} \triangleq r_{\max}/c_{\min}$.

Note this is different with the classical RM method [31, 34] which adopts the samples R_{S_t,a_t} and C_{S_t,a_t} for updating θ_t . Finally, our Robbins-Monro based decision is defined as follows:

$$\arg \max_{a \in \mathcal{A}(S_t)} \hat{r}_{S_t,a} - \theta_t \check{c}_{S_t,a}$$

where θ_t is treated as an estimation of θ^* .

This method circumvents the need to directly estimate the task arrival distribution. Instead, it learns the optimal cumulative ratio by iteratively adjusting θ with the term $\eta(\hat{r}_{S_t,a_t} - \theta_t \check{c}_{S_t,a_t})$. When θ is excessively large, $\eta(\hat{r}_{S_t,a_t} - \theta_t \check{c}_{S_t,a_t})$ tends to be negative, driving a downward shift, whereas a small θ prompts its value to skew positive, inducing an upward adjustment in itself. With a proper

learning rate η , θ can steadily converge to the optimal balance ratio as iterations increase (please refer to Section 4).

4 THEORETICAL RESULTS

To present our main results, we first introduce the common assumptions on rewards and costs.

ASSUMPTION 1. *The reward $R_{S_t,a}$ is a sub-Gaussian random variable with mean $r_{S_t,a} = \mathbb{E}[R_{S_t,a}] \in [r_{\min}, r_{\max}]$ for any $S_t \in \mathcal{S}$, $a \in \mathcal{A}(S_t)$ and $t \in [T]$.*

ASSUMPTION 2. *The cost $C_{S_t,a}$ is a positive sub-Gaussian random variable with mean $c_{S_t,a} = \mathbb{E}[C_{S_t,a}] \in [c_{\min}, c_{\max}]$ for any $S_t \in \mathcal{S}$, $a \in \mathcal{A}(S_t)$ and $t \in [T]$.*

Regret & Convergence Gap: Let θ^* be the optimal reward-to-cost ratio in Problem (1), we define $Gap(T)$ to be the *convergence gap*:

$$Gap(T) \triangleq \left| \theta^* - \frac{\sum_{t=1}^T \mathbb{E}[R_{S_t,a_t}]}{\sum_{t=1}^T \mathbb{E}[C_{S_t,a_t}]} \right|, \quad (10)$$

which measures the distance between the cumulative reward-to-cost ratio returned by a policy and the optimal ratio. We define the *regret* to be:

$$\mathcal{R}(T) \triangleq T \cdot Gap(T).$$

Our goal is to show that DOL-RM achieves sub-linear regret $o(T)$, i.e., $\lim_{T \rightarrow \infty} \frac{\mathcal{R}(T)}{T} = \lim_{T \rightarrow \infty} Gap(T) = 0$, which implies that DOL-RM converges to the optimal policy and achieves the best reward-to-cost ratio in the long-term. We state our main result for DOL-RM in the following theorem.

THEOREM 4.1. *Suppose Assumption 1 and 2 hold, DOL-RM achieves the following regret:*

$$\mathcal{R}(T) = O(T^{\frac{3}{4}}).$$

Theorem 4.1 highlights DOL-RM's favorable theoretical performance in achieving *no regret learning* with sub-linear regrets $O(T^{\frac{3}{4}})$, which implies the convergence gaps of $O(T^{-\frac{1}{4}})$. To the best of our knowledge, these are the first results for online task scheduling problems without any prior information on rewards, costs and task arrival distributions. These results also indicate that DOL-RM can quickly identify an effective and efficient policy that converges to the optimal ratio θ^* with the integral design of double-optimistic learning and the Robbins-Monro method.

We want to further mention two related works cited as [39] and [31]. [39] studied the reward-to-cost ratio in a Markov decision process (MDP). Though MDP includes bandit as a special case, [39] only established an asymmetrical convergence. [31] assumed perfect information on the rewards and costs and established improved regrets of $O(\sqrt{T})$. With such perfect information, [31] only needs to quantify the uncertainty of task arrival without any bias¹ from rewards and costs. This enables [31] to conduct an aggressive learning scheme to control the bias only from the task arrival such that it can provide an anytime gap performance and improved performance. However, when the rewards and costs are unknown, we must carefully balance all these uncertainties and control the

¹Rewards and costs are usually biased with noises, e.g., sub-Gaussian noise in our case, while [31] assumes constant rewards and costs, thus no bias involved.

bias over the entire time horizon (Lemma 4.2 and 4.3), which requires a conservative learning scheme and advanced Lyapunov drift techniques to achieve sub-linear gap given coupled uncertainties.

Next, we present the detailed proof of Theorem 4.1 and focus on the analysis of convergence gap $Gap(T)$, where we first decompose $Gap(T)$ into the two items related to the double-optimistic learning and Robbins-Monro iteration method, respectively, and then established the items individually.

4.1 Proof of Theorem 4.1

Recall the definition of convergence gap in (10). We first decompose the convergence gap using triangle inequality by involving $\frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t,a_t}]}{\sum_{t=1}^T \mathbb{E}[\hat{c}_{S_t,a_t}]}$ as follows

$$Gap(T) \leq \underbrace{\left| \frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t,a_t}]}{\sum_{t=1}^T \mathbb{E}[\hat{c}_{S_t,a_t}]} - \frac{\sum_{t=1}^T \mathbb{E}[r_{S_t,a_t}]}{\sum_{t=1}^T \mathbb{E}[c_{S_t,a_t}]} \right|}_{\text{Double-optimistic learning error}}. \quad (11)$$

$$+ \underbrace{\left| \theta^* - \frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t,a_t}]}{\sum_{t=1}^T \mathbb{E}[\hat{c}_{S_t,a_t}]} \right|}_{\text{Robbins-Monro iteration convergence gap}}. \quad (12)$$

For the first term (11), we establish the bound in Lemma 4.2, which is related to the bias of optimistic learning on rewards and costs. For the second term (12), we establish the bound in Lemma 4.3 by quantifying the cumulative bias from the Robbins-Monro iteration.

LEMMA 4.2. *Suppose Assumption 1 and 2 hold, DOL-RM achieves*

$$\left| \frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t,a_t}]}{\sum_{t=1}^T \mathbb{E}[\hat{c}_{S_t,a_t}]} - \frac{\sum_{t=1}^T \mathbb{E}[r_{S_t,a_t}]}{\sum_{t=1}^T \mathbb{E}[c_{S_t,a_t}]} \right| = O(T^{-\frac{1}{2}}).$$

LEMMA 4.3. *Suppose Assumption 1 and 2 hold, DOL-RM achieves*

$$\left| \theta^* - \frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t,a_t}]}{\sum_{t=1}^T \mathbb{E}[\hat{c}_{S_t,a_t}]} \right| = O(T^{-\frac{1}{4}}).$$

Based on these two lemmas, we prove the convergence gap in Theorem 4.1 as follows:

$$Gap(T) \leq O(T^{-\frac{1}{2}}) + O(T^{-\frac{1}{4}}) = O(T^{-\frac{1}{4}}).$$

Here we only offer order-wise results in $O(\cdot)$ for the sake of presentation and the detailed expressions are delegated to the Appendix.

Next, we prove Lemma 4.2 and 4.3, respectively.

4.2 Double-Optimistic Learning Analysis

Lemma 4.2 represents the estimation error of the cumulative reward-to-cost ratio. Due to the page limit, we illustrate the key steps in the analysis of double-optimistic learning and the completed version can be found in Appendix C.

Recall that to optimize the reward-to-cost ratio given unknown reward and cost functions, we employ optimistic estimators for both rewards and costs. This indicates that the following inequalities hold with a high probability according to UCB/LCU learning [24]

$$0 \leq \hat{r}_{S,a} - r_{S,a} \leq 0, \quad 0 \leq \frac{1}{\hat{c}_{S,a}} - \frac{1}{c_{S,a}} \leq 0.$$

These guarantee optimistic estimation for cumulative terms $\sum_{t=1}^T r_{S_t, a_t}$ and $\frac{1}{\sum_{t=1}^T c_{S_t, a_t}}$, resulting in an optimistic estimation for cumulative reward-to-cost ratio $\frac{\sum_{t=1}^T \mathbb{E}[r_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[c_{S_t, a_t}]}$.

To proceed, we define *partial double-optimistic errors*

$$\begin{aligned}\epsilon(T) &\triangleq \left| \frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}]} - \frac{\sum_{t=1}^T \mathbb{E}[r_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}]} \right|, \\ \kappa(T) &\triangleq \left| \frac{\sum_{t=1}^T \mathbb{E}[r_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}]} - \frac{\sum_{t=1}^T \mathbb{E}[r_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[c_{S_t, a_t}]} \right|.\end{aligned}$$

Such errors directly form an upper bound of the double-optimistic learning error in (11) via the triangle inequality:

$$\left| \frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}]} - \frac{\sum_{t=1}^T \mathbb{E}[r_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[c_{S_t, a_t}]} \right| \leq \epsilon(T) + \kappa(T),$$

We then bound these partial double-optimistic errors as follows:

$$\begin{aligned}\epsilon(T) + \kappa(T) &= \frac{1}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}]} \left| \sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t}] - \sum_{t=1}^T \mathbb{E}[r_{S_t, a_t}] \right| \\ &\quad + \frac{\sum_{t=1}^T \mathbb{E}[r_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}] \sum_{t=1}^T \mathbb{E}[c_{S_t, a_t}]} \left| \sum_{t=1}^T \mathbb{E}[c_{S_t, a_t}] - \sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}] \right| \\ &\leq \frac{1}{T c_{\min}} \left| \sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t} - r_{S_t, a_t}] \right| + \frac{r_{\max}}{T c_{\min}^2} \left| \sum_{t=1}^T \mathbb{E}[c_{S_t, a_t} - \check{c}_{S_t, a_t}] \right|,\end{aligned}$$

where the last inequality holds due to the boundedness of reward and cost. As shown in the above inequality, we break down the estimated error of cumulative ratio into the estimated errors of rewards and costs. Consequently, we only need to bound the following terms

$$\left| \sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t} - r_{S_t, a_t}] \right|, \quad \left| \sum_{t=1}^T \mathbb{E}[c_{S_t, a_t} - \check{c}_{S_t, a_t}] \right|,$$

which have been widely studied in UCB/LCB learning [24, 27, 28] and both are in the order of $O(\sqrt{T})$. Eventually, we have

$$\epsilon(T) + \kappa(T) \leq O(T^{-\frac{1}{2}}),$$

which proves Lemma 4.2.

4.3 Robbins-Monro Iteration Analysis

Lemma 4.3 is the key to establishing the convergence gap and regret for DOL-RM and is also the most challenging part. We first establish its upper bound with the terms related to $\mathbb{E}[(\theta_t - \theta^*)^2]$ in the following Lemma 4.4, we then carefully control the cumulative errors in the Robbins-Monro iteration.

LEMMA 4.4. *Suppose Assumption 1 and 2 hold, under DOL-RM, we bound the Robbins-Monro iteration convergence gap in (12) as follows:*

$$\left| \theta^* - \frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}]} \right| \leq \max \left\{ \left| \frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}]} - \frac{\sum_{t=1}^T \mathbb{E}[r_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[c_{S_t, a_t}]} \right|, \right. \quad (13)$$

$$\left. \frac{\sqrt{2C_1}}{T c_{\min}} \sum_{t=1}^T \sqrt{\mathbb{E}[(\theta_t - \theta^*)^2]} \right\}. \quad (14)$$

We provide the proof sketch by considering two cases: 1) when $\theta^* \leq \frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}]}$, the term in (13) is upper bound of (12) according to the definition of θ^* . 2) when $\frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}]} \leq \theta^*$, we have the greedy decision in our algorithm such that

$$\begin{aligned}\mathbb{E}[\hat{r}_{S_t, a_t} - \theta_t \check{c}_{S_t, a_t} | \theta_t] &\geq \mathbb{E}[R_{S_t, a^*} | \theta_t] - \theta_t \mathbb{E}[C_{S_t, a^*} | \theta_t] \\ &\geq r^* - \theta_t c^*,\end{aligned}$$

where a^* is any decision within $\mathcal{A}(S_t)$, the first inequality holds due to the greedy decision and $\hat{r}_{S_t, a_t} \geq r_{S_t, a_t}$, $\check{c}_{S_t, a_t} \leq c_{S_t, a_t}$ holds with a high probability; the second inequality holds because (r^*, c^*) lies within the closure of available decision set. By adding $(\theta_t - \theta^*)\check{c}_{S_t, a_t}$ on both sides of the above inequality, we further have

$$\mathbb{E}[\hat{r}_{S_t, a_t} - \theta^* \check{c}_{S_t, a_t}] \geq \mathbb{E}[(\theta_t - \theta^*)(\check{c}_{S_t, a_t} - c^*)].$$

Finally, one could use Cauchy-Schwarz inequality to establish (14) in Lemma 4.4. More details can be found in the Appendix A.

Since we already have the $O(T^{-\frac{1}{2}})$ bound of (13) in Section 4.2, the key is to establish the cumulative error

$$\sum_{t=1}^T \sqrt{\mathbb{E}[(\theta_t - \theta^*)^2]}.$$

We leverage the Lyapunov drift analysis [29, 38] to study this key term. Note that [31] also used this technique to study the cumulative error term, where they assumed the perfect information of rewards and costs such that it can establish anytime error of $\mathbb{E}[(\theta_t - \theta^*)^2]$ via one-step Lyapunov drift. However, we do not assume any of such knowledge and quantify the cumulative error term by aggregating the bias over all tasks.

Define the Lyapunov function

$$V_t \triangleq \frac{1}{2}(\theta_t - \theta^*)^2,$$

we analyze the corresponding Lyapunov drift

$$\Delta(t) \triangleq V_{t+1} - V_t$$

and have the following lemma.

LEMMA 4.5. *Under DOL-RM, we have the expected Lyapunov drift to be bounded as follows:*

$$\begin{aligned}\mathbb{E}[\Delta(t)] &\leq -2c_{\min}\eta\mathbb{E}[V_t] + \eta^2 b \\ &\quad + \theta_{\text{gap}}\eta\mathbb{E}[\hat{r}_{S_t, a_t} - r_{S_t, a_t} - \theta^*(\check{c}_{S_t, a_t} - c_{S_t, a_t})],\end{aligned}$$

where $\theta_{\text{gap}} \triangleq (\theta_{\max} - \theta_{\min})$ and $b \geq \frac{1}{2}\mathbb{E}[(\hat{r}_{S_t, a_t} - \theta_t \check{c}_{S_t, a_t})^2], \forall t$.

We in the following provide the proof sketch for this key lemma. According to the reward-to-cost ratio learning in (8), we have

$$\begin{aligned} (\theta_{t+1} - \theta^*)^2 &= \left(\left[\theta_t + \eta(\hat{r}_{S_t, a_t} - \theta_t \check{c}_{S_t, a_t}) \right]_{\theta_{\min}}^{\theta_{\max}} - \theta^* \right)^2 \\ &= \left[\left(\theta_t + \eta(\hat{r}_{S_t, a_t} - \theta_t \check{c}_{S_t, a_t}) - \theta^* \right)_{\theta_{\min}}^{\theta_{\max}} \right]^2 \\ &\leq (\theta_t + \eta(\hat{r}_{S_t, a_t} - \theta_t \check{c}_{S_t, a_t}) - \theta^*)^2. \end{aligned}$$

By the definition of the Lyapunov drift Δ_t , we have

$$\begin{aligned} \mathbb{E}[\Delta(t)] &= \frac{1}{2} \mathbb{E}[(\theta_{t+1} - \theta^*)^2] - \frac{1}{2} \mathbb{E}[(\theta_t - \theta^*)^2] \\ &\leq \eta^2 b + \eta \mathbb{E}[(\theta_t - \theta^*)(\hat{r}_{S_t, a_t} - \theta_t \check{c}_{S_t, a_t})]. \end{aligned}$$

Similar to the proof of Lemma 4.4, we bound the last term in the above inequality under two cases:

1) For $\theta^* \geq \theta_t$, we can directly get

$$\mathbb{E}[(\theta_t - \theta^*)(\hat{r}_{S_t, a_t} - \theta_t \check{c}_{S_t, a_t})] \leq -2c_{\min} \eta \mathbb{E}[V_t],$$

since $\mathbb{E}[\hat{r}_{S_t, a_t} - \theta_t \check{c}_{S_t, a_t} | \theta_t] \geq r^* - \theta_t c^* = c^*(\theta^* - \theta_t)$. Then the inequality in Lemma 4.5 holds by the fact $\mathbb{E}[\hat{r}_{S_t, a_t} - r_{S_t, a_t}] \geq 0$, $\mathbb{E}[c_{S_t, a_t} - \check{c}_{S_t, a_t}] \geq 0$ which is guaranteed by the property of UCB/LCB.

2) For $\theta^* \leq \theta_t$, given the boundedness of cost, we have

$$\begin{aligned} &\mathbb{E}[(\theta_t - \theta^*)(\hat{r}_{S_t, a_t} - \theta_t \check{c}_{S_t, a_t})] \\ &\leq \mathbb{E}[(\theta_t - \theta^*)(\hat{r}_{S_t, a_t} - \theta^* \check{c}_{S_t, a_t})] - c_{\min} \mathbb{E}[(\theta_t - \theta^*)^2]. \end{aligned}$$

We then have the following bound by decomposing the first term and incorporating the fact that $\theta_t - \theta^* \leq \theta_{\text{gap}}$, i.e.,

$$\begin{aligned} &\mathbb{E}[(\theta_t - \theta^*)(\hat{r}_{S_t, a_t} - \theta^* \check{c}_{S_t, a_t})] \\ &\leq \mathbb{E}[(\theta_t - \theta^*)(r_{S_t, a_t} - \theta^* c_{S_t, a_t})] \\ &\quad + \theta_{\text{gap}} \mathbb{E}[(\hat{r}_{S_t, a_t} - r_{S_t, a_t} - \theta^*(\check{c}_{S_t, a_t} - c_{S_t, a_t}))]. \end{aligned}$$

Next, drawing upon the definition of the optimal ratio θ^* , which indicates that $\mathbb{E}[(r_{S_t, a_t} - \theta^* c_{S_t, a_t})] \leq 0$, we complete the proof of Lemma 4.5. More details are provided in Appendix B.

Proving Lemma 4.3: We rearrange the inequality in Lemma 4.5 and take summation from 1 to T over it, yielding the following:

$$\begin{aligned} \sum_{t=1}^T \mathbb{E}[V_t] &\leq - \sum_{t=1}^T \frac{\mathbb{E}[\Delta(t)]}{2c_{\min} \eta} + \frac{T \eta b}{2c_{\min}} \\ &\quad + \frac{\theta_{\text{gap}}}{2c_{\min}} \sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t} - r_{S_t, a_t} - \theta^*(\check{c}_{S_t, a_t} - c_{S_t, a_t})] \\ &\leq \frac{V_1}{2c_{\min} \eta} + \frac{T \eta b}{2c_{\min}} + O(\sqrt{T}) = O(\sqrt{T}), \end{aligned}$$

where the second inequality holds because of $\sum_{t=1}^T \Delta(t) = V_T - V_1$ and the fact that the estimation errors are at the order of $O(\sqrt{T})$ according to UCB/LCB learning (similarly stated in Section 4.2); the last equality holds by plugging the learning rate $\eta = O(1/\sqrt{T})$.

According to the Cauchy-Schwarz inequality, we have,

$$\sum_{t=1}^T \sqrt{\mathbb{E}[V_t]} \leq \sqrt{T \sum_{t=1}^T \mathbb{E}[V_t]},$$

which implies that the key cumulative error is at the order of $O(T^{\frac{3}{4}})$ in the following lemma.

LEMMA 4.6. *Under DOL-RM, we have*

$$\sum_{t=1}^T \sqrt{\mathbb{E}[(\theta_t - \theta^*)^2]} = O(T^{\frac{3}{4}}).$$

Finally, by combining Lemma 4.2 and 4.6 into Lemma 4.4, we establish the Robbins-Monro iteration convergence gap as follows:

$$\left| \theta^* - \frac{\sum_{t=1}^T \mathbb{E}[\hat{r}_{S_t, a_t}]}{\sum_{t=1}^T \mathbb{E}[\check{c}_{S_t, a_t}]} \right| \leq \max \left\{ O(T^{-1/2}), O(T^{-1/4}) \right\},$$

which proves Lemma 4.3.

5 NUMERICAL RESULTS

We evaluate DOL-RM via both synthetic simulation and real-world experiment in terms of the cumulative reward-to-cost ratio $\frac{\sum_{t=1}^T R_{S_t, a_t}}{\sum_{t=1}^T C_{S_t, a_t}}$. For both cases, we let learning rate $\eta_t = 1/c_{\min}(t+1)$ in our DOL-RM algorithm.

Baselines: We consider the following representative baselines to justify DOL-RM's performance:

- General Thompson Sampling (TS) [35].
- Classic Upper Confidence Bound (UCB) algorithm [24].
- ω -UCB [19], a variance-aware cost-efficient algorithm.
- Oracle algorithm [31], which needs prior knowledge of reward and cost and an optimal solution².

5.1 Synthetic Simulation

Two-Task-Type We consider a synthetic setting similar to the toy example presented in Figure 1. We consider two incoming types of tasks with arrival rates of $\{p, 1-p\}$ and reward-cost vectors of $\{(3, 1), (3, 2), (1, 1)\}$. The observations $R_{S, a}$ and $C_{S, a}$ are corrupted with additive Gaussian noise sampled from $\mathcal{N}(0, 1)$.

We let p and $1-p$ be the arrival probability of task S_x and S_y , respectively. To verify the robustness of DOL-RM against varied task arrival distributions, we test it under two task arrival patterns, quantified by $p = 0.8$, and $p = 0.6$ in Figure 4(a) and 4(b).

Seven-Task-Type We consider a more complex task scheduling problem with 7 types of incoming tasks with arrival rates of $\{0.3, 0.1, 0.2, 0.1, 0.05, 0.1, 0.15\}$ and reward-cost vectors of $\{(3, 1), (3, 2), (1, 1), (2, 1), (2.5, 1.5), (2, 1), (1, 1), (3, 2), (1.5, 1.5), (2.5, 1)\}$. The observations $R_{S, a}$ and $C_{S, a}$ are corrupted with additive Gaussian noise sampled from $\mathcal{N}(0, 1)$.

Figure 4 demonstrates that DOL-RM outperforms the state-of-the-art learning-based algorithms in terms of the cumulative reward-to-cost ratio. DOL-RM's superior performance is consistent across two different cases, showcasing its high adaptability. Compared to the oracle algorithm [31], which has full prior knowledge and the optimal ratio in hindsight, DOL-RM exhibits a fast convergence rate, indicating its efficiency in identifying the optimal policy under certainty.

²We adopt this algorithm as an ideal benchmark as it possesses full knowledge before decision-making, allowing us to verify whether DOL-RM converges to the optimal ratio. However, it cannot be applied in the real world due to the lack of prior knowledge.

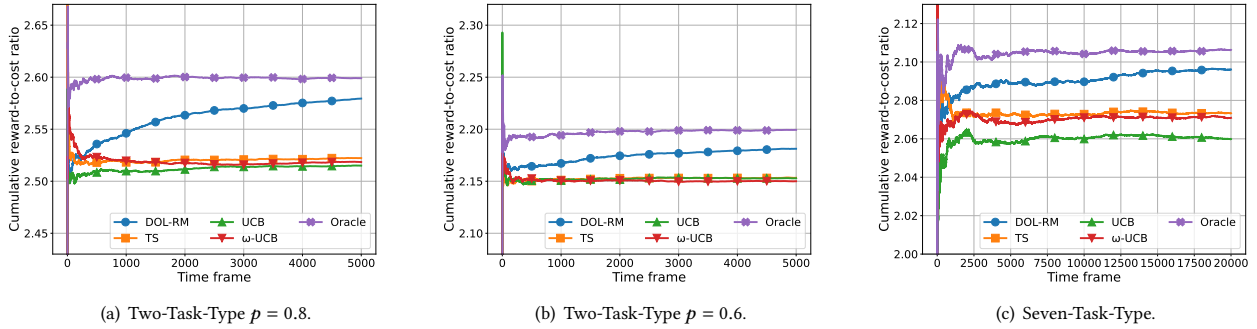


Figure 4: Synthetic Experiment: Performance Comparison in a Two-Task-Type Case and Seven-Task-Type Case.

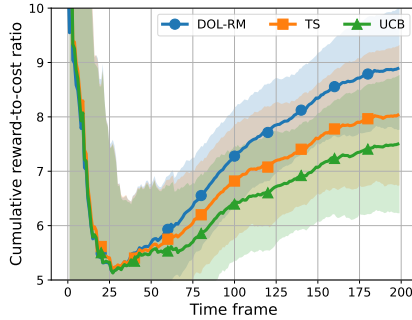


Figure 5: Real-World Experiment.

5.2 Real-World Experiment

We apply DOL-RM to schedule machine learning (ML) training tasks in a shared server. Specifically, we consider five types of ML classification tasks in the system, where a satellite image classification task S_s with four categories, a weather classification task S_w and a rice classification task S_r with five categories, a natural scene classification task S_n with six categories and a cat & dog classification task S_c . Let $\mathcal{S} = \{S_s, S_w, S_r, S_n, S_c\}$. For each classification task, the server decides a training epoch from a range of options. The available options on epoch number are within $\mathcal{A}(S_s) = \mathcal{A}(S_r) = \mathcal{A}(S_c) = \{1, 2, 3, 4, 5\}$ and $\mathcal{A}(S_w) = \mathcal{A}(S_n) = \{1, 5, 10, 15, 20\}$ ³, respectively. After training the model with an epoch number $a_t \in \mathcal{A}(S_t)$ for a classification task $S_t \in \mathcal{S}$, we can obtain the AUC on the test set as reward R_{S_t, a_t} and observe the training time as cost C_{S_t, a_t} ⁴.

The dataset of satellite image classification has more than 1000 pictures in each category of satellite remote-sensing images [33]. All the pictures are resized into 255×255 and 20% of them are separated as the test set. The dataset of weather classification contains boasting over 300 pictures of weather images in each category [18]. All the pictures are resized into 150×150 and we construct the

³For satellite image classification S_s , each epoch contains 141 steps. For weather classification S_w , each epoch contains 38 steps. For rice classification S_r , each epoch contains 100 steps. For natural scene classification S_n , each epoch contains 150 steps. For cat & dog classification S_c , each epoch has 30 steps.

⁴We conduct min-max normalization on reward and cost to facilitate practical training.

test set with 20% of the pictures. We use the first 1000 pictures of rice images in each class in [23]. The pictures are all resized into 224×224 and 20% of them are reserved as the test set. We take the first 500 pictures in each category in the training set of [4] as training data and utilize the test set of [4] as test data. All the pictures are resized into 150×150 . The database of cat & dog classification contains 11875 pictures for each class of cat or dog [36], where the pictures are also scaled into size 150×150 and 10% of these pictures are reserved as the test set. Our experiment was conducted on an RTX 3080 Ti GPU, running a 64-bit Ubuntu 18.04 system. The detailed structures and parameters of neural network models can be found in Appendix E.

Let the arrival probability of $\{S_s, S_w, S_r, S_n, S_c\}$ be $\{0.1, 0.4, 0.2, 0.2, 0.1\}$. As ω -UCB [19] and oracle algorithm [31] requires non-causal information, which is infeasible in the experiment, we compare DOL-RM only with the (classical) UCB and TS. Figure 5 plot the cumulative reward-to-cost ratio for these algorithms where the light-shaded areas indicate the corresponding standard deviation. These results show DOL-RM outperforms the baselines significantly and demonstrate that DOL-RM can converge to a better policy in the real-world system even without any prior information.

6 CONCLUSION

In this paper, we initiated the study on a novel formulation of the online task scheduling problem, where the task arrival distribution, rewards, and costs are all unknown. Guided by double-optimistic learning and Robbins-Monro method, we proposed an effective and efficient algorithm DOL-RM which integrates optimistic estimations and stochastic approximation of balancing point. We theoretically demonstrated its superior performance with a simultaneous achievement of sub-linear regret bound and fast learning. Via justification in synthetic and real-world scenarios, we not only showed the outperformance of DOL-RM over state-of-the-art baselines but also envisioned enormous potential applications of our modeling and algorithm design.

ACKNOWLEDGMENTS

The work was partly supported by the Shanghai Sailing Program 22YF1428500, the National Nature Science Foundation of China under grant 62302305. Corresponding author: Xin Liu.

REFERENCES

- [1] Dr Amit Agarwal and Saloni Jain. 2014. Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment. *International Journal of Computer Trends and Technology* 9, 7 (2014), 344–349.
- [2] Afra A Alabadi and Maysoun F Abulkhair. 2021. Multi-Objective Task Scheduling Optimization in Spatial Crowdsourcing. *Algorithms* 14, 3 (2021), 77–97.
- [3] AR Arunarani, Dhanabalachandran Manjula, and Vijayan Sugumaran. 2019. Task Scheduling Techniques in Cloud Computing: A Literature Survey. *Future Generation Computer Systems* 91 (2019), 407–415.
- [4] PUNEET BANSAL. 2019. *Intel Image Classification*. Retrieved December 12, 2023 from <https://www.kaggle.com/datasets/puneet6060/intel-image-classification/data>
- [5] Semih Cayci, Atilla Eryilmaz, and Rayadurgam Srikant. 2019. Learning to Control Renewal Processes with Bandit Feedback. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3, 2 (2019), 1–32.
- [6] Semih Cayci, Atilla Eryilmaz, and R Srikant. 2020. Budget-Constrained Bandits over General Cost and Reward Distributions. In *Proceedings of AISTATS*.
- [7] Duo Cheng, Ruiquan Huang, Cong Shen, and Jing Yang. 2022. Cascading Bandits with Two-Level Feedback. In *Proceedings of IEEE ISIT*.
- [8] Debojit Das, Shweta Jain, and Sujit Gujar. 2022. Budgeted Combinatorial Multi-Armed Bandits. *arXiv preprint arXiv:2202.03704* (2022).
- [9] Dingxiong Deng, Cyrus Shahabi, and Linhong Zhu. 2015. Task Matching and Scheduling for Multiple Workers in Spatial Crowdsourcing. In *Proceedings of ACM SIGSPATIAL*.
- [10] Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu. 2013. Multi-Armed Bandit with Budget Constraint and Variable Costs. In *Proceedings of AAAI*.
- [11] Maciej Drozdowski. 1996. Scheduling Multiprocessor Tasks: An Overview. *European Journal of Operational Research* 94, 2 (1996), 215–230.
- [12] Bennett Fox. 1966. Markov Renewal Programming by Linear Fractional Programming. *SIAM J. Appl. Math.* 14, 6 (1966), 1418–1432.
- [13] Chao Gan, Ruida Zhou, Jing Yang, and Cong Shen. 2018. Cost-Aware Learning and Optimization for Opportunistic Spectrum Access. *IEEE Transactions on Cognitive Communications and Networking* 5, 1 (2018), 15–27.
- [14] Chao Gan, Ruida Zhou, Jing Yang, and Cong Shen. 2020. Cost-Aware Cascading Bandits. *IEEE Transactions on Signal Processing* 68 (2020), 3692–3706.
- [15] Chao Gao, Tong Mo, Taylor Zowtuk, Tanvir Sajed, Laiyuan Gong, Hanxuan Chen, Shangling Jui, and Wei Lu. 2021. Bansor: Improving Tensor Program Auto-Scheduling with Bandit Based Reinforcement Learning. In *Proceedings of IEEE ICTAI*.
- [16] Lizheng Guo, Shuguang Zhao, Shigen Shen, and Changyuan Jiang. 2012. Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm. *Journal of Networks* 7, 3 (2012), 547–553.
- [17] Sachi Gupta, Vikas Kumar, and Gaurav Agarwal. 2010. Task Scheduling in Multiprocessor System using Genetic Algorithm. In *Proceedings of IEEE ICMLC*.
- [18] VIJAY GUPTA. 2019. *Weather Classification*. Retrieved July 13, 2023 from <https://www.kaggle.com/datasets/vijaygiitk/multiclass-weather-dataset>
- [19] Marco Heyden, Vadim Arzamasov, Edouard Fouché, and Klemens Böhm. 2023. Budgeted Multi-Armed Bandits with Asymmetric Confidence Intervals. *arXiv preprint arXiv:2306.07071* (2023).
- [20] Mohammad Hossin and Md Nasir Sulaiman. 2015. A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process* 5, 2 (2015), 01–11.
- [21] Cengiz Kahraman, Orhan Engin, Ihsan Kaya, and R Elif Öztürk. 2010. Multi-processor Task Scheduling in Multistage Hybrid Flow-Shops: A Parallel Greedy Algorithm Approach. *Applied Soft Computing* 10, 4 (2010), 1293–1300.
- [22] Roman Khazankin, Harald Psailer, Daniel Schall, and Schahram Dustdar. 2011. QoS-Based Task Scheduling in Crowdsourcing Environments. In *Proceedings of ICSC*.
- [23] MURAT KOKLU. 2022. *Rice Image Dataset*. Retrieved December 12, 2023 from <https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset/data>
- [24] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit Algorithms*. Cambridge University Press.
- [25] Haifang Li and Yingce Xia. 2017. Infinitely Many-Armed Bandits with Budget Constraints. In *Proceedings of AAAI*.
- [26] Rui Li, Qian Ma, Jie Gong, Zhi Zhou, and Xu Chen. 2021. Age of processing: Age-Driven Status Sampling and Processing Offloading for Edge-Computing-Enabled Real-Time IoT Applications. *IEEE Internet of Things Journal* 8, 19 (2021), 14471–14484.
- [27] Xin Liu, Bin Li, Pengyi Shi, and Lei Ying. 2020. Pond: Pessimistic-Optimistic Online Dispatching. *arXiv preprint arXiv:2010.09995* (2020).
- [28] Xin Liu, Bin Li, Pengyi Shi, and Lei Ying. 2021. An Efficient Pessimistic-Optimistic Algorithm for Stochastic Linear Bandits with General Constraints. In *Advances in Neural Information Processing Systems 34 - 35th Conference on Neural Information Processing Systems, NeurIPS 2021*.
- [29] Michael J Neely. 2010. Stochastic Network Optimization with Application to Communication and Queueing Systems. *Synthesis Lectures on Communication Networks* 3, 1 (2010), 1–211.
- [30] Michael J. Neely. 2013. Dynamic Optimization and Learning for Renewal Systems. *IEEE Trans. Automat. Control* 58, 1 (2013), 32–46.
- [31] Michael J Neely. 2021. Fast Learning for Renewal Optimization in Online Task Scheduling. *Journal of Machine Learning Research* 22, 1 (2021), 12785–12828.
- [32] Fahimeh Ramezani, Jie Lu, Javid Taheri, and Farookh Khader Hussain. 2015. Evolutionary Algorithm-Based Multi-Objective Task Scheduling Optimization Model in Cloud Environments. *World Wide Web* 18, 6 (2015), 1737–1757.
- [33] MAHMOUD REDA. 2021. *Satellite Image Classification*. Retrieved December 12, 2023 from <https://www.kaggle.com/datasets/mahmouredda55/satellite-image-classification/data>
- [34] Herbert Robbins and Sutton Monro. 1951. A Stochastic Approximation Method. *The Annals of Mathematical Statistics* 22, 3 (1951), 400–407.
- [35] Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. 2018. A Tutorial on Thompson Sampling. *Foundations and Trends® in Machine Learning* 11, 1 (2018), 1–96.
- [36] SACHIN. 2019. *Cats-vs-Dogs*. Retrieved July 15, 2023 from <https://www.kaggle.com/datasets/shaunthesheep/microsoft-catsvsdogs-dataset>
- [37] Anirban Santara, Gaurav Aggarwal, Shuai Li, and Claudio Gentile. 2022. Learning to Plan Variable Length Sequences of Actions with a Cascading Bandit Click Model of User Feedback. In *Proceedings of AISTATS*.
- [38] Rayadurgam Srikant and Lei Ying. 2014. *Communication Networks: An Optimization, Control, and Stochastic Networks Perspective*. Cambridge University Press.
- [39] Wesley Suttle, Kaiqing Zhang, Zhuoran Yang, Ji Liu, and David Kraemer. 2021. Reinforcement Learning for Cost-Aware Markov Decision Processes. In *Proceedings of ICML*.
- [40] Long Tran-Thanh, Archie Chapman, Enrique Munoz De Cote, Alex Rogers, and Nicholas R Jennings. 2010. Epsilon-First Policies for Budget-Limited Multi-Armed Bandits. In *Proceedings of AAAI*.
- [41] Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas Jennings. 2012. Knapsack Based Optimal Policies for Budget-Limited Multi-Armed Bandits. In *Proceedings of AAAI*.
- [42] Chao Wang, Ruida Zhou, Jing Yang, and Cong Shen. 2019. A Cascading Bandit Approach To Efficient Mobility Management In Ultra-Dense Networks. In *Proceedings of International Workshop on IEEE MLSP*.
- [43] Yingce Xia, Wenkui Ding, Xu-Dong Zhang, Nenghai Yu, and Tao Qin. 2016. Budgeted Bandit Problems with Continuous Random Costs. In *Proceedings of ACML*.
- [44] Yingce Xia, Tao Qin, Wenkui Ding, Haifang Li, Xudong Zhang, Nenghai Yu, and Tie-Yan Liu. 2017. Finite Budget Analysis of Multi-Armed Bandit Problems. *Neurocomputing* 258 (2017), 13–29.
- [45] Datong Zhou and Claire Tomlin. 2018. Budget-Constrained Multi-Armed Bandits with Multiple Plays. In *Proceedings of AAAI*.