# Learning-Aided Online Task Offloading for UAVs-Aided IoT Systems

Junge Zhu, Xi Huang, Yinxu Tang, Ziyu Shao
School of Information Science and Technology, ShanghaiTech University
Email: {zhujg, huangxi, tangyx, shaozy}@shanghaitech.edu.cn

***Abstract*—Equipped with specific IoT on-board devices, unmanned aerial vehicles (UAVs) can be orchestrated to assist in particular value-added service delivery with improved quality-of-service. Typically, services are delegated in the unit of tasks to a designated *leader UAV*, while the leader UAV splits each task into sub-tasks and offloads them to part of its nearby UAVs, *a.k.a. helper UAVs*, for timely processing. Such a decision making process, often referred to as *UAV task offloading*, still remains open and challenging to design, due to various uncertainties therein, such as the resource availability and instant workloads on helper UAVs. However, existing solutions often assume the knowledge of system dynamics is fully available and conduct decision making in an offline manner, resulting in excessive control overheads and scalability issues. In this paper, we study the UAV task offloading problem in an online setting and formulate it as a multi-armed bandits (MAB) problem with time-varying resource constraints. Then we propose *VR-LATOS*, a learning-aided offloading scheme that learns the unknown statistics from feedback signals while making effective offloading decisions in an online fashion. Results from both theoretical analysis and simulations demonstrate that *VR-LATOS* outperforms state-of-the-art schemes.**

## I. Introduction

By extending *unmanned aerial vehicles* (UAVs) with IoT devices, the emergence of UAVs-aided IoT platforms in recent years has driven the development of various IoT value-added services, ranging from surveillance, agriculture services, disaster relief to military services [1]–[3].

Typically, in an UAVs-aided IoT system, some on-demand value-added services [4] [5] may be delegated to a cluster of orchestrated UAVs, each equipped with resource-constrained IoT on-board devices, so as to mitigate the workloads in some hot-spot areas and deliver better quality-of-service. The period during which UAVs flock together to carry out service delivery is often referred to as the *link expiration time* [6]. Within this period, one of the UAVs is designated as the *leader UAV* and the rest as *helper UAVs*. The leader UAV constantly receives tasks of particular services that demand timely processing, decomposes each task into sub-tasks depending on their statefulness, and offloads some sub-tasks to part of the helper UAVs over wireless channels, *a.k.a. UAV task offloading*.

Despite the potential benefits of task offloading in UAVs-aided IoT systems, it still remains non-trivial and challenging to design an effective offloading scheme under the following considerations. To ensure the timely processing of tasks, it is favorable for the leader UAV to offload sub-tasks onto helper UAVs which have idle and adequate computational resources. However, helper UAVs' dynamics are usually time-varying and not instantly accessible to the leader UAV, because 1) the helper UAVs may be busy processing some local tasks in the mean time, and 2) sub-tasks vary in processing times. Instead, they can only be obtained or inferred from feedback signals after sub-tasks' completion. One of the consequences of such uncertainties is that, if sub-tasks are offloaded to helper UAVs with inadequate computational resources, they may be delayed or even discarded, causing longer task latency and degrading quality-of-service. Therefore, the offloading scheme must be able to *learn* system dynamics from as much feedback as possible by helper UAVs while exploiting such information effectively to guide subsequent decision making, achieving an *exploitation-and-exploration tradeoff*. Meanwhile, the decision making must be performed in a computational efficient and timely manner, since high computational complexity or excessive communication overheads may incur extra latency, offsetting the benefits brought by offloading.

Recent works have proposed various schemes to apply UAV offloading in different scenarios. For example, Ouahouah *et al.* [6] studied UAV task offloading problem and formulated the problem as two integer programming problems with objectives in latency reduction and energy efficiency, respectively. Lyu *et al.* [7] introduced a new hybrid network architecture for cellular systems by leveraging UAVs for data offloading. Then they presented a spectrum-reuse scheme to maximize the minimum throughput. Hu *et al.* [8] explored the optimal spectrum trading contract design for UAV-assisted 5G networks, and further proposed a dynamic programming algorithm for UAV offloading. All the above work assumes the availability of full knowledge of system dynamics and conduct UAV offloading in an offline manner. However, there is no empirical evidence showing that information about system dynamics can be attained in advance.

In this paper, we consider the UAV task offloading problem in an online setting, where the leader UAV, with online task arrivals, makes offloading decisions while requiring no *a priori* information about system dynamics but only cumulative feedback signals from helper UAVs, to maximize the availability probability of chosen helper UAVs with time-varying resource constraints. Our contributions are summarized as follows:

- ◇ **Problem Formulation:** We are the first to consider the UAV task offloading problem in an online setting. Specifically, we formulate the problem as a multi-armed bandits problem with time-varying resource constraints.
- ◇ **Algorithm Design and Performance Analysis:** By

adapting recently developed network application optimization techniques [9] and applying biased estimators, we propose Variance-Reduced Learning-Aided Task Offloading Scheme (*VR-LATOS*), which learns unknown system statistics through feedback and conducts effective decision making in a distributed manner. Our theoretical analysis shows that VR-LATOS achieves sub-linear regret and violation.

- ◇ **Experimental Verification:** We conduct simulations to evaluate VR-LATOS. Results show that VR-LATOS outperforms state-of-the-art schemes with notable reduction in cumulative reward variance.

The rest of the paper is organized as follows. Section II presents the system model and problem formulation. Section III elaborates the algorithm design, followed by its performance analysis. Section IV shows the simulation results, while Section V concludes the paper.

## II. System Model and Problem Formulation

In this section, we introduce the system model and then formulate the UAV task offloading problem as a stochastic constrained multi-armed bandits (MAB) problem with time-varying resource constraints. We summarize the key notations in Table I.

TABLE I: Key Notations

| Notation | Description |
|---|---|
| $U_0$ | The leader UAV |
| $\mathcal{U}$ | Set of helper UAVs, including $\{U_1, \ldots, U_K\}$ |
| $K$ | Number of helper UAVs |
| $k$ | Number of selected helper UAVs |
| $A_i(t)$ | Random variable of availability state of helper UAV $U_i$ in time slot $t$ |
| $B_i(t)$ | Random variable of computational resources of helper UAV $U_i$ in time slot $t$ |
| $C_i(t)$ | Compound available computational resources of helper UAV $U_i$ in time slot $t$ |
| $D_i(t)$ | Random variable of latency signal from helper UAV $U_i$ in time slot $t$ ($i = 1, 2, \ldots, K$) |
| $a_i^t$ | Realization value of $A_i(t)$ |
| $b_i^t$ | Realization value of $B_i(t)$ |
| $c_i^t$ | Realization value of normalized $C_i(t)$ |
| $d_i^t$ | Realization value of $D_i(t)$ |
| $a_i$ | Mean of availability probability of helper UAV $U_i$ |
| $p_i^t$ | Probability of selecting helper UAV $U_i$ in time slot $t$ |
| $\mathcal{L}_t$ | Set of selected helper UAVs in time slot $t$ |

### A. System Model

We consider an UAVs-aided IoT platform consisting of a cluster of UAVs to deliver some delegated value-added services. Each UAV is equipped with on-board IoT devices and communicates with other UAVs through wireless connections. Amongst such UAVs are a *leader UAV*, denoted by $U_0$, and $K$ *helper UAVs*, denoted by set $\mathcal{U} \triangleq \{U_1, U_2, \ldots, U_K\}$. During their link expiration time (sliced into $T$ time slots), the leader UAV $U_0$ constantly receives new tasks, one in each time slot $t = 1, \ldots, T$. To be processed timely, each newly arriving task is decomposed into a set of parallel sub-tasks, some to be treated locally by $U_0$ and some to be offloaded to other helper UAVs. For those to be offloaded, we assume that each task contains exactly $k$ ($1 \leq k \leq K$) sub-tasks to be offloaded; meanwhile, each sub-task has the same size of $s_t$ and can be treated by any helper UAVs in $\mathcal{U}$. [1] We denote such an offloading decision by $\mathcal{L}_t \subseteq \mathcal{U}$ such that $|\mathcal{L}_t| = k$.

In the course of decision making, to finish tasks in a timely fashion, some system dynamics must be taken into account.

On one hand, it is favorable for the leader UAV $U_0$ to offload sub-tasks to helper UAVs with idle computational resources. This is because some helper UAVs may be busy processing their own tasks and hence not available to accept the offloaded sub-task; such rejection incurs unnecessary latency, thereby delaying the sub-task completion. We quantify such availability by associating each $U_i \in \mathcal{U}$ with a binary availability state $a_i^t$ for each time slot $t$, such that $a_i^t = 1$ indicates $U_i$ is available and 0 otherwise. Each $a_i^t$ is assumed to be independent across helper UAVs and sampled from a Bernoulli distribution with constant mean $a_i$. We denote $(a_1, \ldots, a_K)$ by $\boldsymbol{a}$ and $(a_1^t, \ldots, a_K^t)$ by $\boldsymbol{a}_t$.

On the other hand, helper UAVs often vary in their computational resources and hence processing capacities; therefore, it is also advisable to offload sub-tasks to those with more computational resources, so that they can be finished more quickly. We denote the amounts of helper UAV $U_i$'s computational resource, *e.g.*, number of CPU cycles, in time $t$ by $b_i^t$. For notational simplicity, we denote $\boldsymbol{b}_t = (b_1^t, \ldots, b_K^t)$.

### B. Problem Formulation

With the above model and considerations, for leader UAV $U_0$, its goal is to find an effective offloading policy $\boldsymbol{\pi}$ that induces timely processing of sub-tasks over $T$ time slots. In fact, one can derive an optimal policy with full knowledge of system dynamics. However, in practice, such a priori information, including $\boldsymbol{a}_t$, $\boldsymbol{b}_t$, $\boldsymbol{c}_t$, $\boldsymbol{a}$, is often not available and hence unknown to $U_0$. For example, the availability of helper UAVs, denoted by $\boldsymbol{a}_t$ in time slot $t$ is only revealed after sub-tasks' completion through feedback sent to the leader UAV $U_0$. In face of these uncertainties, the offloading policy is expected to be able to *learn* the unknown statistics through as much feedback as possible from different helper UAVs, while effectively exploiting attained feedback to make best possible decisions within finite time slots, *a.k.a.*, the *exploitation-exploration tradeoff*. Such a design is non-trivial and challenging.

Inspired by recent work on learning-aided network application optimization [9], we formulate the UAV task offloading problem as a stochastic multi-armed bandits (MAB) problem with time-varying resource constraints. Typically, a MAB

[1] Our model can also be extended to the scenario where different tasks vary in the number of offloaded sub-tasks and sub-tasks vary in sizes.

problem considers the interaction between an agent and its environment through rounds of independent decision making. In each round, the agent executes an action from a given set to the environment. Based on the chosen action, the environment will reveal a reward to the agent, which is sampled from an unknown distribution (different across actions). The agent's goal is to derive an optimal policy that decides an action sequence to achieve the maximum possible cumulative rewards. As for MAB with multi-level reward signals and reward guarantee, it extends the original setting by allowing the agent to choose more than one but a fix number of actions in each round, receive and learn the compound of various reward signals, and ensure the average reward signals stay above some predetermined threshold.

To recast our model into this setting, we view the leader UAV $U_0$ as the agent and the helper UAVs in $\mathcal{U}$ as the action set. The goal of the agent is thus to choose $k$ actions (helper UAVs) out of $\mathcal{U}$. To be explorative, we require the policy $\boldsymbol{\pi}$ to be stochastic, by introducing $\boldsymbol{p}_t^{\pi} \triangleq (p_1^t, \ldots, p_K^t)$, a selection probability vector for each time slot $t$, where $p_i^t \in [0,1]$ denotes the probability of UAV $U_i$ being chosen.

When it comes to reward signals, remind that the agent's goal is to choose helper UAVs with both high availability probability and sufficient computational resources. The possible choices of reward signals are $\boldsymbol{a}_t$ and $\boldsymbol{b}_t$. Since they are unknown to the agent ($U_0$) before sub-tasks' completion, we define $A_i(t)$ as the random variable for the availability signal of helper UAV $U_i$ in time slot $t$ and its realization is $a_i^t$. Remind that $a_i^t$ follows some stationary distribution with mean $a_i$ and hence $\mathrm{E}\{A_i(t)\} = a_i$. Note that $a_i$ can also be viewed as the availability probability of helper UAV $U_i$. Likewise, we denote by $B_i(t)$ as the random variable for the amounts of computational resource on helper UAV $U_i$ in time slot $t$ with its realization as $b_i^t$. In practice, however, the signal $B_i(t)$ is often not directly attainable; instead, it can be inferred from the latency signals fed back after sub-tasks' completion. We denote the latency signal from $U_i$ in time $t$ by random variable $D_i(t)$ with realization $d_i^t$. Considering resource constraints and the proximity of the agent and helper UAVs, we assume that their transmission time in between is negligible compared to the processing time of sub-tasks. Thus we can write each sub-task's (with size $s_t$) latency as $D_i(t) = s_t / B_i(t)$. However, if $U_i$ is not available in time slot $t$, *i.e.*, $a_i^t = 0$, then the sub-task will be discarded and no signal $D_i(t)$ will be fed back. In such a case, we set $D_i(t) = +\infty$. Next, by defining the compound reward signal as $C_i(t) \triangleq A_i(t)/D_i(t) = A_i(t) \cdot B_i(t)/s_t$, the agent's goal is also equivalent to choose UAVs with high availability probability and short latency. To minimize the impact of variance in sub-task sizes, we normalize $\tilde{C}_i(t) = C_i(t) \cdot s_t$ and denote its realization vector in time $t$ by $\boldsymbol{c}_t$.

In addition, considering that quite a few IoT services require real-time control [3], the leader UAV $U_0$ should also try to prevent the processing of sub-tasks from being discarded by UAVs with heavy workloads. To this end, we require that the time-average availability of helper UAVs chosen by offloading decisions be above some constant threshold $h$, *i.e.*, $\boldsymbol{a}^T \boldsymbol{p}_t \geq h$.

Consequently, we define the UAV task offloading problem as follows,

$$\begin{aligned} \underset{[\boldsymbol{p}_t^{\pi}]_t}{\text{Maximize}} \quad & \mathbb{E}\left\{\sum_{t=1}^{T} \boldsymbol{c}_t^T \boldsymbol{p}_t^{\pi}\right\} \\ \text{Subject to} \quad & \boldsymbol{a}^T \boldsymbol{p}_t \geq h. \end{aligned} \tag{1}$$

Given full knowledge of system dynamics, an oracle policy $\boldsymbol{\pi}^*$ can be derived such that it always makes the best offloading decisions to maximize $\sum_t \boldsymbol{c}_t^T \boldsymbol{p}_t$ without violating $\boldsymbol{a}^T \boldsymbol{p}_t \geq h$. However, as mentioned before, such information is usually not accessible in practice. Therefore, we switch to design a stochastic policy $\boldsymbol{\pi}$ that approaches the performance of the oracle policy as close as possible. To measure the performance gap between policy $\boldsymbol{\pi}$ and oracle $\boldsymbol{\pi}^*$, we adopt two criteria, including cumulative regret, defined as

$$R_{\pi}(T) = \max_{\boldsymbol{a}^T \boldsymbol{p}_t \geq h} \sum_{t=1}^{T} \boldsymbol{c}_t^T \boldsymbol{p}_t - \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{c}_t^T \boldsymbol{p}_t^{\pi}\right], \tag{2}$$

and cumulative violation, defined as

$$V_{\pi}(T) = \mathbb{E}\left[\sum_{t=1}^{T} (h - \boldsymbol{a}^T \boldsymbol{p}_t^{\pi})\right]^+. \tag{3}$$

Note that small regret implies the policy $\pi$ gets closer to the oracle policy. On the other hand, small violation indicates that UAVs which are chosen to process the offloaded sub-tasks are available most of time. Both compound reward signals and the number of available helper UAVs contribute to the timely processing of tasks. However, for helper UAVs with considerable computational resources and low availability probabilities, the two reward signals may be conflicting. To avoid selecting helper UAVs with low availability probabilities, which will lead to large latency, the policy $\pi$ should take both the regret and violation into account.

## III. Algorithm Design & Performance Analysis

In this section, we elaborate the design of the learning-aided task offloading scheme, followed by its performance analysis.

### A. Algorithm Design

The main challenge of designing policy $\boldsymbol{\pi}$ lies in how to balance the tradeoff between maximizing the cumulative compound rewards and maintaining low violation as defined in (3). To address the challenge, we adapt recently developed network application optimization techniques [9] to design the UAV task offloading scheme. By employing an adjustable coefficient, *i.e.*, the Lagrange multiplier $l_t$, to balance the tradeoff between regret and violation, and two biased estimators to estimate the reward distribution, we propose *VR-LATOS*, *i.e.*, *Variance-Reduced Learning-Aided Task Offloading Scheme*, and show its pseudocode in Algorithm 1.

Under VR-LATOS, in each time slot $t$, the leader UAV $U_0$ maintains a weight vector $\boldsymbol{w_t} = \{w_1^t, \ldots, w_K^t\}$ for each UAV. With $\boldsymbol{w_t}$, $U_0$ calculates the probability selection vector $\tilde{\boldsymbol{p}}_t = \{p_1^t, \ldots, p_K^t\}$, with each $p_i^t$ corresponding to the probability

**Algorithm 1** Variance-Reduced Learning-Aided Task Offloading Scheme (VR-LATOS)

**Input:** $w^1 \leftarrow \mathbf{1}$, $v \leftarrow (1/k - \lambda/K)/(1-\lambda)$, $l_1 \leftarrow 0$, $\beta \leftarrow \lambda\delta k/(\delta+k)K$.
**Output:** Action sequence $\{\mathcal{L}_t\}_{t=1}^T$.

1: **for** $t$ in $\{1, ..., T\}$ **do**
2: $\mathcal{X}_t \leftarrow \emptyset$, $\mathcal{L}_t \leftarrow \emptyset$.
3: **if** $\max_{i\in\{1,...,K\}} w_i^t \geq v \sum_{i=1}^K w_i^t$ **then**
4: Solve $u_t$ such that $u_t/(\sum_{i=1, w_i^t \geq u_t}^K u_t + \sum_{i=1, w_i^t < u_t}^K w_i^t) = v$.
5: Set $\mathcal{X}_t \leftarrow \{i : w_i^t \geq u_t\}$.
6: **for** $i$ in $\{1, ..., K\}$ **do** $\tilde{w}_i^t \leftarrow u_t$ if $i \in \mathcal{X}_t$; otherwise $\tilde{w}_i^t \leftarrow w_i^t$.
7: **for** $i$ in $\{1, ..., K\}$ **do** $\tilde{p}_i^t \leftarrow k[(1-\lambda)\tilde{w}_i^t / \sum_{i=1}^K \tilde{w}_i^t + \lambda/K]$.
8: $\mathcal{L}_t \leftarrow \text{DepRound}(k, \tilde{\boldsymbol{p}}_t)$.
9: **for** $i \in \mathcal{L}_t$ **do** receive $a_i^t$ and $b_i^t$.
10: **for** $i$ in $\{1, ..., K\}$ **do** $\hat{a}_i^t \leftarrow a_i^t \mathbf{1}(i \in \mathcal{L}_t)/\tilde{p}_i^t + \theta/\tilde{p}_i^t$, $\hat{c}_i^t \leftarrow a_i^t b_i^t \mathbf{1}(i \in \mathcal{L}_t)/\tilde{p}_i^t + \theta/\tilde{p}_i^t$.
11: **for** $i$ in $\{1, ..., K\}$ **do**

$$w_i^{t+1} \leftarrow \begin{cases} w_i^t & \text{if } i \in \mathcal{X}_t, \\ w_i^t \exp[\beta(\hat{c}_i^t + l_t \hat{a}_i^t)] & \text{if } i \notin \mathcal{X}_t. \end{cases}$$

12: $l_{t+1} \leftarrow [(1-\delta\beta)l_t - \beta(\frac{\hat{\boldsymbol{a}}_t^T \tilde{\boldsymbol{p}}_t}{1-\lambda} - h)]^+$.

**function** DepRound($k$, $\boldsymbol{p}$)
**while** exist $i \wedge p_i \in (0,1)$ **do**
Find $i, j, i \neq j$, such that $p_{i,j} \in (0,1)$.
$m \leftarrow \min\{1-p_i, p_j\}$; $n \leftarrow \min\{p_i, 1-p_j\}$.

$$(p_i, p_j) \leftarrow \begin{cases} (p_i + m, p_j - m) \text{ with prob. } \frac{n}{m+n}, \\ (p_i - n, p_j + n) \text{ with prob. } \frac{m}{m+n}. \end{cases}$$

**return** $\mathcal{L} \leftarrow \{1 \leq i \leq K \,|\, p_i = 1\}$

of helper UAV $U_i$ being chosen. Specifically, from line 3 - 7, the leader UAV $U_0$ calculates the probability selection vector $\tilde{\boldsymbol{p}}_t$ using $\boldsymbol{w_t}$, where line 3 - 6 ensure the probabilities in $\boldsymbol{p_t}$ no greater than 1. In line 8, by applying the dependent rounding function (*DepRound*) with $\tilde{\boldsymbol{p}}_t$, $U_0$ selects $k$ helper UAVs to offload to. After sub-tasks' processing, UAV $U_0$ obtains the reward signals, $a_i^t$, $b_i^t$, with respect to each of the selected helper UAVs. Finally, at the end of time slot $t$, $U_0$ updates the weight vector $\boldsymbol{w_t}$ using the reward signals. Note that

$$\hat{a}_i^t = a_i^t \mathbf{1}(i \in \mathcal{L}_t)/\tilde{p}_i^t + \theta/\tilde{p}_i^t, \tag{4}$$

$$\hat{c}_i^t = a_i^t b_i^t \mathbf{1}(i \in \mathcal{L}_t)/\tilde{p}_i^t + \theta/\tilde{p}_i^t \tag{5}$$

are biased estimators for $a_i^t$ and $c_i^t$, since

$$\mathbb{E}(\hat{a}_i^t) = a_i^t + \theta/\tilde{p}_i^t > a_i^t \text{ and } \mathbb{E}(\hat{c}_i^t) = c_i^t + \theta/\tilde{p}_i^t > c_i^t.$$

According to (4) and (5), the introduction of the biased estimates induces more explorative decision making. Specifically, some helper UAVs may have a high time-average reward but have been underestimated due to cumulative rewards so far. In such a case, the second term $\theta/\tilde{p}_i^t$ offsets the inferiority of such helper UAVs and leaves them extra chance to be chosen.

Next, we unfold *DepRound* in detail. Basically, *DepRound* probabilistically updates $\tilde{\boldsymbol{p}}_t$ until $\tilde{p}_i^t$ is either 0 or 1, while satisfying the condition that $\mathbf{1}^T \tilde{\boldsymbol{p}}_t = k$. Then $U_0$ would choose $k$ UAVs whose $\tilde{p}_i^t$ is 1. At line 9, VR-LATOS receives the rewards $a_i^t$ and $b_i^t$, and then estimates $a_i$ and $c_i$ by yielding estimates $\hat{a}_i^t$ and $\hat{c}_i^t$ in line 10. Finally, the weight vector $\boldsymbol{w_t}$ and the Lagrange multiplier $l_t$ are updated, respectively, as shown in line 11 and 12.

In addition, we note that VR-LATOS can be run in an computationally-efficient manner in each time slot $t$. Particularly in Algorithm 1, line 4 requires a $K$-round enumeration in the worst case, while for the rest for-loops, including line 6 - 11, each of them requires $K$ iteration and each iteration requires constant complexity. Hence, the computational complexity for VR-LATOS is $O(K)$, linear in the number of helper UAVs, implying that VR-LATOS can make offloading decisions with low overheads.

### B. Performance Analysis

We show that *VR-LATOS* achieves both a sub-linear regret and violation bounds, as speficied by the following theorem.

**Theorem 1.** *Let* $\zeta \triangleq \frac{\gamma\eta k}{(\eta+k)K}, \gamma \triangleq \min\left(1, \sqrt{\frac{2(e-2)K+Kk}{k\ln(K/k)T^{2/3}}}\right)$ *and* $\eta \triangleq \frac{4(e-2)\gamma k}{1-\gamma}$. *Over* $T$ *time slots, VR-LATOS achieves sub-linear bounds for both regret and violation, as follows:*

$$R_{\tilde{\pi}}(T) \leq O(kK\ln(K)T^{\frac{2}{3}}) \tag{6}$$

*and*

$$V_{\tilde{\pi}}(T) \leq O(k^{\frac{1}{2}}K^{\frac{1}{2}}T^{\frac{5}{6}}), \tag{7}$$

*where* $R_\pi(\cdot)$ *and* $V_\pi(\cdot)$ *are defined in (2) and (3), respectively.*

The proof is delegated to Appendix-A. We also note that VR-LATOS achieves the same regret and violation bounds as LMG [9]. However, the introduction of biased estimators in VR-LATOS conduces to further reduction in the variance of compound rewards, resulting in stationary resource usage and less energy consumption, which are important for energy-limited UAVs to extend their lifetime. Besides, smaller variance implies more effective learning process, which often leads to high availability of selected helper UAVs and hence shorter latency of tasks. See more details in Section IV.

## IV. Simulation

In this section, we present simulation results to verify the effectiveness of *VR-LATOS* while comparing it against two state-of-the-art schemes, *CUCB* [10] and *LMG* [9].

### A. Basic Settings

We conduct the simulation in an UAVs-aided IoT platform with 11 UAVs, where one of them is fixed as $U_0$ with $K = 10$ and $k = 3$. To eliminate the impact of randomness, we repeat 50 simulations and each simulation is run over $T = 1 \times 10^4$

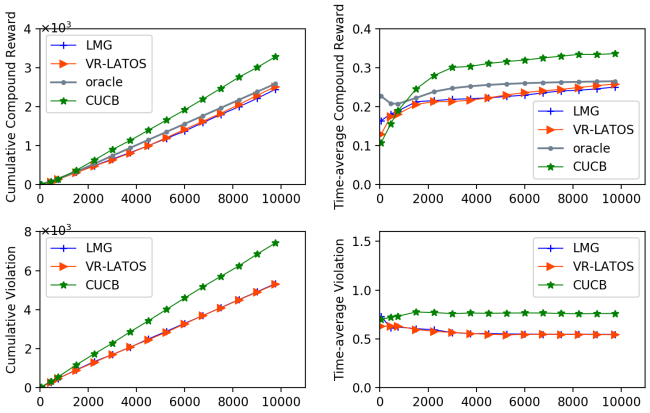


Fig. 1: Comparison of compound rewards and violations.

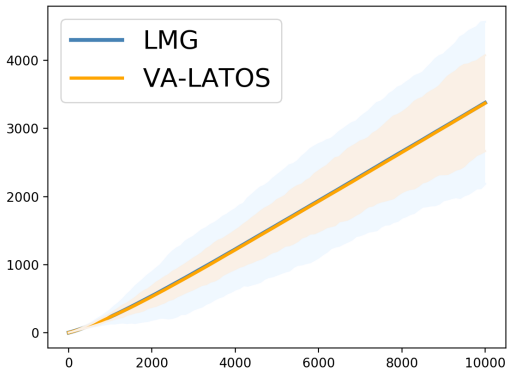


Fig. 2: Variances of VR-LATOS and LMG

time slots. We set the threshold $h = 2$. The availability states $\boldsymbol{a}_t$ are Bernoulli random variables, with its mean $a_i$ as the availability probabilities of UAVs. Then the availability state equals 1 if the UAV is available and 0 otherwise. We generate the values of computational resources uniformly from $[0, b_i]$, where $b_i$ is the total computational resources of $U_i$. Both $a_i$ and $b_i$ are uniformly generated from $[0, 1]$.

### *B. Simulation Results*

**Reward and Violation:** We compare VR-LATOS with CUCB and LMG. CUCB always selects the top-3 helper UAVs with the highest UCB (upper confidence bound) indices $\bar{c}_i^t + \sqrt{3 \ln t/(2N_i(t))}$, where $\bar{c}_i^t$ is the estimate of the compound reward and $N_i(t)$ is the number of times that arm $i$ has been selected by the time slot $t$. LMG algorithm use the unbiased estimates $\hat{a}_i^t = a_i^t/p_i^t \mathbf{1}(i \in \mathcal{L}_t), \hat{c}_i^t = a_i^t b_i^t/p_i^t \mathbf{1}(i \in \mathcal{L}_t)$. We also implement the oracle policy. Since oracle knows $\boldsymbol{a}$ and $\boldsymbol{b}_t$ in each time $t$, it can calculate the optimal selection vector $\boldsymbol{p}_t^*$ by solving $\max_{\boldsymbol{a}^T \boldsymbol{p} \geq h} (\boldsymbol{a} \circ \boldsymbol{b}_t)^T \boldsymbol{p}$ in each time slot $t$, where $\circ$ is the element-wise product operator between vectors.

We compare the cumulative compound reward over time slots, time-average compound reward, cumulative violation over time slots, and time-average violation of VR-LATOS, LMG, and CUCB. Note that we don't compare the regret because the definition of the regret of VR-LATOS and LMG is different from CUCB: the regret of VR-LATOS and LMG is defined in (2) which compares the reward with the constrained optimal policy while the regret of CUCB compares the reward with unconstrained optimal policy. In addition, as shown in (2), since oracle is an optimal and deterministic policy, its reward is a constant, so minimizing the regret and maximizing the reward are totally equivalent. Specifically, the cumulative compound reward by time slot $t$ is calculated by $\sum_{t'=1}^{t} \sum_{i \in \mathcal{L}_t'} c_i^{t'}$. The cumulative compound reward for oracle is calculated by $\sum_{t'=1}^{t} (\boldsymbol{a} \circ \boldsymbol{b}_t')^T \boldsymbol{p}_{t'}^*$. The cumulative violation at $t$ is calculated by $\sum_{t'=1}^{t} (h - \sum_{i \in \mathcal{L}_t'} a_i^{t'})$. The results are shown in Figure 1. We can see that the cumulative rewards of VR-LATOS and LMG are very close to oracle. As $t$ increases, the time-average compound rewards of VR-LATOS and LMG increase and get closer to oracle. The cumulative compound and time-average compound reward of CUCB are larger than VR-LATOS and LMG. The reason is that the objective of CUCB is maximizing the cumulative reward without considering the constraint. The cumulative violations of VR-LATOS and LMG are smaller than CUCB, and the time-average violations of VR-LATOS and LMG keep decreasing. This also means VR-LATOS and LMG can learn the knowledge and have more accurate estimation of the availability probabilities and computational resources of the helper UAVs. The simulation results show that VR-LATOS and LMG are effective in UAV task offloading problem.

**Mean and Variance:** From Figure 1 we can see that the performance of VR-LATOS and LMG are very close, including the reward and the violation. By introducing biased estimators, the reward of VR-LATOS is well concentrated about its mean and the variance is small. In this experiment, we compare the variance of rewards of VR-LATOS and LMG.

Figure 2 shows the curves of the expectations of rewards induced by VR-LATOS and LMG. Besides, the colored areas show their variances, respectively. We see that their expectations are very close. However, the variance of VR-LATOS is smaller than LMG. As mentioned in III, variance reduction leads to less energy consumption and smaller latency. Therefore, VR-LATOS is more effective than LMG in UAV task offloading problem.

## V. Conclusion

In this paper, we studied the problem of task offloading in UAVs-aided IoT platform. By formulating the problem as a multi-armed bandits problem with time-varying multi-level reward signals and reward guarantee, we proposed *VR-LATOS*, a learning-aided scheme which derives a stochastic policy that induces UAV task offloading decisions in an online fashion. Results from both theoretical analysis and simulations show that *VR-LATOS* achieves sub-linear regret and violation bounds, while outperforming state-of-the-art schemes

by achieving a notable reduction in the cumulative reward variance.

## REFERENCES


[1] M. Asadpour, B. Van den Bergh, D. Giustiniano, K. A. Hummel, S. Pollin, and B. Plattner, "Micro aerial vehicle networks: An experimental analysis of challenges and opportunities," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 141–149, 2014.

[2] N. H. Motlagh, M. Bagaa, and T. Taleb, "Uav-based iot platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[3] H. Shakhatreh, A. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles: A survey on civil applications and key research challenges," *arXiv preprint arXiv:1805.00881*, 2018.

[4] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of unmanned aerial vehicles*. Springer, 2015.

[5] N. H. Motlagh, M. Bagaa, and T. Taleb, "Uav selection for a uav-based integrative iot platform," in *Proceedings of IEEE Globecom*, 2016.

[6] S. Ouahouah, T. Taleb, J. Song, and C. Benzaid, "Efficient offloading mechanism for uavs-based value added services," in *Proceedings of IEEE ICC*, 2017.

[7] J. Lyu, Y. Zeng, and R. Zhang, "Uav-aided offloading for cellular hotspot," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3988–4001, 2018.

[8] Z. Hu, Z. Zheng, L. Song, T. Wang, and X. Li, "Uav offloading: Spectrum trading contract design for uav assisted 5g networks," *arXiv preprint arXiv:1801.05388*, 2017.

[9] K. Cai, X. Liu, Y.-Z. J. Chen, and J. C. Lui, "An online learning approach to network application optimization with guarantee," in *Proceedings of IEEE INFOCOM*, 2018.

[10] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *Proceedings of ICML*, 2013.


## APPENDIX A
## PROOF OF THEOREM 1

From line 12 of the algorithm we have

$$l_{t+1}=\left[(1-\delta\beta)l_t-\beta\left(\frac{\hat{\boldsymbol{a}}_t{}^\top\tilde{\boldsymbol{p}}_t}{1-\lambda}-h\right)\right]^+\leq[(1-\delta\beta)l_t+\beta h)]^+. \tag{8}$$

In (1) inequation, $\tilde{\boldsymbol{p}}_t$ represents is the probabilistic selection vector of the policy $\tilde{\pi}$ at the time $t$.

By induction on $l_t$, we construct

$$l_{t+1}-\frac{h}{\delta}\leq(1-\delta\beta)(l_t-\frac{h}{\delta}). \tag{9}$$

It is obvious that $\{l_t-\frac{h}{\delta}\}$ is geometric series. We obtain $l_t\leq\frac{h}{\delta}$ from $l_t-\frac{h}{\delta}\leq(1-\delta\beta)^{t-1}(l_1-\frac{h}{\delta})$. Let $\Phi_t=\sum_{i=1}^M w_i^t$ and $\tilde{\Phi}_t=\sum_{i=1}^M\tilde{w}_i^t$. Define $\boldsymbol{r}_t\triangleq\boldsymbol{c}_t+l_t\boldsymbol{a}_t$ and $\hat{\boldsymbol{r}}_t\triangleq\hat{\boldsymbol{c}}_t+l_t\hat{\boldsymbol{a}}_t$. Let $\boldsymbol{p}_t$ be an arbitrary probabilistic selection vector which satisfies $p_i^t\in[0,1],\mathbf{1}^\top\boldsymbol{p}_t=k$ and $\boldsymbol{a}^\top\boldsymbol{p}_t\geq h$.

Since $\ln x$ is a concave function, according to Jensen's inequality, $\ln(\frac{x_1+....+x_k}{k})\geq\frac{\ln x_1+\ln x_2+....\ln x_k}{k}$, we obtain that $\ln(x_1+....+x_k)-\ln k\geq\frac{\ln x_1+\ln x_2+....\ln x_k}{k}$.For the sequence of selected $\mathcal{I}_t$ at $t=1,\ldots,T$,

$$\begin{aligned}\sum_{t=1}^T\ln\frac{\Phi_{t+1}}{\Phi_t}&=\ln\frac{\Phi_{T+1}}{\Phi_1}=\ln\left(\sum_{i=1}^K w_i^{T+1}\right)-\ln K\\&\geq\ln\left(\sum_{i=1}^K p_i^t w_i^{T+1}\right)-\ln K\\&\geq\sum_{i=1}^K\frac{p_i^t}{k}\sum_{t:i\notin\mathcal{X}_t}\beta\hat{r}_i^t-\ln\frac{K}{k}\\&=\frac{\beta}{k}\sum_{i=1}^k p_i^t\sum_{t:i\notin\mathcal{X}_t}\hat{r}_i^t-\ln\frac{K}{k}.\end{aligned} \tag{10}$$

As $\beta=\frac{\lambda\delta k}{(\delta+k)K}$ and $l_t\leq\frac{h}{\delta},\beta\hat{r}_i^t\leq1$, we have

$$\begin{aligned}\frac{\Phi_{t+1}}{\Phi_t}&=\sum_{i\in\mathcal{K}/\mathcal{X}_t}\frac{w_i^{t+1}}{\Phi_t}+\sum_{i\in\mathcal{X}_t}\frac{w_i^{t+1}}{\Phi_t}=\sum\frac{w_i^t}{\Phi_t}\exp\left(\beta\hat{r}_i^t\right)+\sum_{i\in\mathcal{X}_t}\frac{w_i^t}{\Phi_t}\\&\leq\sum_{i\in\mathcal{K}/\mathcal{X}_t}\frac{w_i^t}{\Phi_t}\left[1+\beta\hat{r}_i^t+(e-2)\beta^2\left(\hat{r}_i^t\right)^2\right]+\sum_{i\in\mathcal{X}_t}\frac{w_i^t}{\Phi_t}\\&=1+\frac{\tilde{\Phi}_t}{\Phi_t}\sum_{i\in\mathcal{K}/\mathcal{X}_t}\frac{w_i^t}{\tilde{\Phi}_t}\left[\beta\hat{r}_i^t+(e-2)\beta^2\left(\hat{r}_i^t\right)^2\right]\\&\leq1+\frac{\beta}{k(1-\lambda)}\sum_{i\in\mathcal{K}/\mathcal{X}_t}\tilde{p}_i^t\hat{r}_i^t+\frac{(e-2)\beta^2}{k(1-\lambda)}\sum_{i\in\mathcal{K}/\mathcal{X}_t}\tilde{p}_i^t\left(\hat{r}_i^t\right)^2\\&\leq1+\frac{\beta}{k(1-\lambda)}\sum_{i\in\mathcal{K}/\mathcal{X}_t}\tilde{p}_i^t\hat{r}_i^t+\frac{(e-2)\beta^2}{k(1-\lambda)}\sum_{i=1}^K(1+l_t)\,\hat{r}_i^t.\end{aligned} \tag{11}$$

Because $e^y\leq1+y+(e-2)y^2$ for $y\leq1$ and the fact that $\tilde{p}_i^t\hat{r}_i^t=r_i^t\leq1+l_t$ for $i\in I_t$ and $\tilde{p}_i^t\hat{r}_i^t=0$ for $i\notin I_t$ , the (4) inequality holds.

Since $\ln(1+y)\leq y$ for $y\geq0$, we have

$$\ln\frac{\Phi_{t+1}}{\Phi_t}\leq\frac{\beta}{k(1-\lambda)}\sum_{i\in\mathcal{K}/\mathcal{X}_t}\tilde{p}_i^t\hat{r}_i^t+\frac{(e-2)\beta^2}{k(1-\lambda)}\sum_{i=1}^K(1+l_t)\,\hat{r}_i^t.$$

Then using (3) inequality, it follows that

$$\begin{aligned}\frac{\beta}{k}\sum_{i=1}^K p_i^t\sum_{t:i\notin\mathcal{X}_t}\hat{r}_i^t-\ln\frac{K}{k}\leq\frac{\beta}{k(1-\lambda)}\sum_{t=1}^T\sum_{i\in\mathcal{M}/\mathcal{X}_t}\tilde{p}_i^t\hat{r}_i^t\\+\frac{(e-2)\beta^2}{k(1-\lambda)}\sum_{t=1}^T\sum_{i=1}^K(1+l_t)\,\hat{r}_i^t.\end{aligned} \tag{12}$$

As $\tilde{p}_i^t=1$ for $i\in\mathcal{X}_t$, and that

$$\sum_{i=1}^K p_i^t\sum_{t:i\in\mathcal{X}_t}\hat{r}_i^t\leq\frac{1}{1-\lambda}\sum_{t=1}^T\sum_{i\in\mathcal{X}_t}\hat{r}_i^t$$

trivially holds, we have

$$\sum_{t=1}^T\hat{\boldsymbol{r}}_t^\top\boldsymbol{p}_t-\frac{k}{\beta}\ln\frac{K}{k}\leq\frac{\sum_{t=1}^T\hat{\boldsymbol{r}}_t^\top\tilde{\boldsymbol{p}}_t}{1-\lambda}+\frac{(e-2)\beta}{1-\lambda}\sum_{t=1}^T\sum_{i=1}^K(1+l_t)\,\hat{r}_i^t. \tag{13}$$

Taking expectation on both sides, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\hat{\boldsymbol{r}}_t^\top \boldsymbol{p}_t - \frac{1}{1-\lambda}\sum_{t=1}^{T}\hat{\boldsymbol{r}}_t^\top \tilde{\boldsymbol{p}}_t\right] \\
&\le \frac{k}{\beta}\ln\frac{K}{k} + \frac{(e-2)\beta}{1-\lambda}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{i=1}^{K}(1+l_t)\,\hat{r}_i^t\right] \qquad (14)\\
&\le \frac{k}{\beta}\ln\frac{K}{k} + \frac{2(e-2)\beta K}{1-\lambda}T + \frac{2(e-2)\beta K}{1-\lambda}\sum_{t=1}^{T} l_t^2.
\end{aligned}$$

where (7) is from the inequality that

$$\mathbb{E}\left[\sum_{i=1}^{K}(1+l_t)\,\hat{r}_i^t\right] = \sum_{i=1}^{K}(1+l_t)\left(c_i^t + l_t a_i^t\right) \le 2K + 2Kl_t^2. \qquad (15)$$

Next, we define a series of function that

$$f_t(l) \triangleq \frac{\delta}{2}l^2 + l\left(\frac{1}{1-\lambda}\hat{\boldsymbol{a}}_t^{\mathrm{T}}\tilde{\boldsymbol{p}}_t - h\right) \qquad (16)$$

and we have $l_{t+1} = [l_t - \beta\nabla f_t(l_t)]_+$. It is clear that $f_t(\cdot)$ is a convex function for all $t$. Thus, for an arbitrary $l$, we have

$$\begin{aligned}
(l_{t+1}-l)^2 &= \left([l_t - \beta\nabla f_t(l_t)]_+ - l\right)^2 \\
&\le (l_t - l)^2 + 2\beta^2 h^2 + 2\beta^2\frac{\left(\hat{\boldsymbol{a}}_t^\top\tilde{\boldsymbol{p}}_t\right)^2}{(1-\lambda)^2} \qquad (17)\\
&\quad + 2\beta\left[f_t(l) - f_t(l_t)\right].
\end{aligned}$$

Let $\Delta = \left[(l_t-l)^2 - (l_{t+1}-l)^2\right]/(2\beta) + \beta k^2$, so we have

$$\begin{aligned}
f_t(l_t) - f_t(l) &\le \Delta + \frac{\beta\left(\hat{\boldsymbol{a}}_t^\top\tilde{\boldsymbol{p}}_t\right)^2}{(1-\lambda)^2} = \Delta + \frac{\beta k^2\left(\frac{1}{k}\hat{\boldsymbol{a}}_t^\top\tilde{\boldsymbol{p}}_t\right)^2}{(1-\lambda)^2} \\
&\le \Delta + \frac{\beta k^2}{(1-\lambda)^2 k}\sum_{i=1}^{K}\left(\tilde{p}_i^t\hat{a}_i^t\right)^2 \\
&\le \Delta + \frac{\beta k}{(1-\lambda)^2}\sum_{i=1}^{K} a_i^t.
\end{aligned} \qquad (18)$$

Taking expectation over $\sum_{t=1}^{T}\left[f_t(l_t) - f_t(l)\right]$, we have

$$\begin{aligned}
&\mathbb{E}\left[\frac{\delta}{2}\sum_{t=1}^{T}l_t^2 - \frac{\delta}{2}l^2T + \sum_{t=1}^{T}l_t\left(\frac{\hat{\boldsymbol{a}}_t^\top\tilde{\boldsymbol{p}}_t}{1-\lambda} - h\right)\right. \\
&\left. -l\sum_{t=1}^{T}\left(\frac{\hat{\boldsymbol{a}}_t^\top\tilde{\boldsymbol{p}}_t}{1-\lambda} - h\right)\right] \le \frac{l^2}{2\beta} + \beta k^2 T + \frac{\beta kK}{(1-\lambda)^2}T. \qquad (19)
\end{aligned}$$

Combining (7) and (12), we have

$$\begin{aligned}
&\sum_{t=1}^{T}\boldsymbol{c}_t^\top\boldsymbol{p}_t - \frac{\mathbb{E}\left[\sum_{t=1}^{T}\boldsymbol{c}_t^\top\tilde{\boldsymbol{p}}_t\right]}{1-\gamma} + \mathbb{E}\left[-\left(\frac{\eta T}{2} + \frac{1}{2\zeta}\right)\lambda^2\right. \\
&\left. +\lambda\sum_{t=1}^{T}\left(h - \frac{\boldsymbol{a}^\top\tilde{\boldsymbol{p}}_t}{1-\gamma}\right)\right] \le \frac{k}{\zeta}\ln\frac{K}{k} + \frac{2(e-2)\zeta KT}{1-\gamma} \\
&+\zeta k^2 T + \frac{\zeta kKT}{(1-\gamma)^2} + \left(\frac{2(e-2)\zeta K}{1-\gamma} - \frac{\eta}{2}\right)\sum_{t=1}^{T}\lambda_t^2 \\
&+\mathbb{E}\left[\sum_{t=1}^{T}\lambda_t\left(h - \boldsymbol{a}^\top\boldsymbol{p}_t\right)\right]. \qquad (20)
\end{aligned}$$

Since $\beta = \frac{\lambda\delta k}{(\delta+k)K}$ and $\delta \ge \frac{4(e-2)\lambda k}{1-\lambda} - k$, we have $\frac{2(e-2)\beta K}{1-\lambda} \le \frac{\delta}{2}$. As $a^\top p_t \ge h$, we have

$$\begin{aligned}
&(1-\lambda)\sum_{t=1}^{T}\boldsymbol{c}_t^\top\boldsymbol{p}_t - \mathbb{E}\left[\sum_{t=1}^{T}\boldsymbol{c}_t^{\mathrm{T}}\tilde{\boldsymbol{p}}_t\right] \\
&+\mathbb{E}\left[l\sum_{t=1}^{T}\left((1-\lambda)h - \boldsymbol{c}^{\mathrm{T}}\tilde{\boldsymbol{p}}_t\right) - \left(\frac{\delta T}{2} + \frac{1}{2\beta}\right)l^2\right] \\
&\le \frac{k}{\beta}\ln\frac{K}{k} + 2(e-2)\beta KT + \beta k^2 T + \frac{\beta kKT}{1-\lambda}. \qquad (21)
\end{aligned}$$

Let $l = \frac{\sum_{t=1}^{T}\left((1-\lambda)h - \boldsymbol{a}^\top\tilde{\boldsymbol{p}}_t\right)}{\delta T + 1/\beta}$. Maximize over $\boldsymbol{p}_t$ and we have,

$$\begin{aligned}
&\max_{\boldsymbol{a}^\top\boldsymbol{p}_t \ge h}\sum_{t=1}^{T}\boldsymbol{c}_t^{\mathrm{T}}\boldsymbol{p}_t - \mathbb{E}\left[\sum_{t=1}^{T}\boldsymbol{c}_t^\top\tilde{\boldsymbol{p}}_t\right] \\
&+\mathbb{E}\left\{\frac{\left[\sum_{t=1}^{T}\left((1-\lambda)h - \boldsymbol{a}^\top\tilde{\boldsymbol{p}}_t\right)\right]^+}{2(\delta T + 1/\beta)}\right\} \le F(T) \qquad (22)
\end{aligned}$$

and $F(T) = \frac{k}{\beta}\ln\frac{K}{k} + 2(e-2)\beta KT + \beta k^2 T + \frac{\beta kKT}{1-\lambda} + \lambda kT$. Then we have results in the form of equation:

$$\mathrm{R}_{\tilde{\pi}}(T) \le F(T),$$

and

$$\mathbf{V}_{\tilde{\pi}}(T) \le \sqrt{2(F(T) + kT)(\delta T + 1/\beta)}.$$

As $\lambda = \min\left(1, \sqrt{\frac{2(e-2)K + Kk}{k\ln(K/k)T^{2/3}}}\right) = \Theta\left(T^{-\frac{1}{3}}\right)$ and $\delta = \frac{4(e-2)\lambda k}{1-\lambda} = \Theta\left(T^{-\frac{1}{3}}\right)$, we have $\beta = \Theta\left(\frac{1}{K}T^{-\frac{2}{3}}\right)$. Finally, we have

$$\mathrm{R}_{\tilde{\pi}}(T) \le O\left(kK\ln(K)T^{\frac{2}{3}}\right)$$

and

$$\mathbf{V}_{\tilde{\pi}}(T) \le O\left(k^{\frac{1}{2}}K^{\frac{1}{2}}T^{\frac{5}{6}}\right).$$

■