IEEE INFOCOM 2026 - IEEE Conference on Computer Communications | 979-8-3315-4961-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/INFOCOM59046.2026.11571205

Constraint-Aware Combinatorial Bandits: Theoretical Foundations and Network Applications

Xiangxiang Dai[1], Jin Li[2], Xutong Liu[3*], Anqi Yu[4], John C.S. Lui[1]
[1]The Chinese University of Hong Kong, [2]Southeast University,
[3]University of Washington, [4]Huawei Technologies Co., Ltd.
Email: {xxdai23, cslui}@cse.cuhk.edu.hk, jin_li@seu.edu.cn, xutongl@uw.edu, yuanqi4@huawei.com

***Abstract*—We study the problem of constraint-aware combinatorial multi-armed bandits (CMAB), a novel extension of combinatorial semi-bandits that maximizes total rewards while adhering to long-term cost constraints. In each round, the environment generates stochastic rewards and costs for each arm from *unknown* distributions. The learning agent selects a combinatorial set of arms that collectively gain rewards and incur costs, which are (partially) observed as feedback to improve future decisions. While prior work has focused on minimizing regret, it often overlooks cost constraints and diverse reward structures across applications. To address this, we introduce a general framework, constraint-aware CMAB with versatile reward functions (`C2MAB-V`). Unlike existing works that solve a difficult discrete constrained optimization (CO) problem to control constraint violation, we propose bypassing the original discrete CO problem and using a novel relaxation and rounding (RR) approach, which solves a relaxed continuous CO problem with improved approximation guarantees and computational efficiency. A new challenge arises since RR is originally designed for offline CO problems, and due to the flexibility of choosing the relaxation function F and the rounding procedure σ, it is unclear how to guarantee that RR can achieve low regret and low violation simultaneously. In this paper, we are the first to discover *unified* RR conditions and, equipped with these conditions, we prove that `C2MAB-V` has the following nice properties: *sub-linear* regret, *fast diminishing* violation, and *good* computational efficiency. We demonstrate the generality of our RR conditions by applying `C2MAB-V` to diverse applications with concrete (F, σ), i.e., mobile crowdsensing, network routing, and multi-LLM selection, with varying reward functions (e.g., linear, conjunctive, disjunctive, submodular) and feedback models (e.g., semi-bandit, cascading). Extensive experiments on synthetic and real-world datasets, including evaluations with nine LLMs on practical tasks, confirm that `C2MAB-V` outperforms baselines across these applications.**

I. Introduction

Stochastic Multi-Armed Bandit (MAB) [1] is a classical framework for sequential decision-making, widely applied in network resource allocation and optimization. It is modeled as a T-round game between a learning agent and L arms, each with rewards drawn from unknown distributions, where $T, L \in \mathbb{N}^+$. In each round, the agent selects an arm, observes a reward, and uses feedback to refine future selections. The goal is to minimize *regret*, the difference between the rewards of always choosing the optimal arm and the agent's policy. Combinatorial MAB (CMAB) extends MAB to scenarios where the agent selects a subset of arms, termed a *super-arm* or *action*, under combinatorial constraints, critical for network applications like routing, load balancing, and resource allocation [2], [3], [4]. CMAB faces versatile reward feedback types: *bandit* (only super-arm reward observed), *semi-bandit* (individual arm outcomes revealed), and *partial* (limited arm outcomes based on a stopping criterion). CMAB must address both the exploration-exploitation tradeoff and the exponential action space, particularly challenging in network optimization where decisions involve paths or resource sets. Besides receiving rewards, it is common that pulling an arm (or a super-arm) will incur different costs (e.g., energy usage, resource consumption, and risk taking). Due to resources/budget limitations or safety restrictions, real-world network applications have long-term cost constraints/requirements so that the T-round averaged costs should not exceed a certain threshold.

To address these challenges, we first extend the CMAB framework by introducing a constraint-aware CMAB problem. In each round, the environment generates a stochastic reward and a stochastic cost for each arm from unknown distributions. The agent selects a super-arm, incurring rewards and costs, and receives semi-bandit or cascading feedback to guide future decisions. Constraint satisfaction is measured by the violation at time T, defined as the T-round average cost minus a predefined threshold b [5]. The agent's objective is to devise a learning policy that maximizes total reward while keeping the average cost below b. This introduces an exploration-exploitation-violation tradeoff: the agent must explore unknown reward and cost parameters while selecting actions that balance reward maximization (i.e., minimizing regret) and cost constraint satisfaction (i.e., minimizing violation).

Second, to tackle this new exploration-exploitation-violation tradeoff, we propose a general framework, constraint-aware CMAB with versatile reward functions (`C2MAB-V`). `C2MAB-V` adopts the Optimism in the Face of Uncertainty (OFU) principle, using the upper confidence bound (UCB) and the lower confidence bound (LCB) to estimate the unknown reward- and cost-related parameters, respectively. A naive approach to action selection involves solving a *discrete* constrained optimization (CO) problem that maximizes the total UCB while ensuring the total LCB remains below a threshold. However, this CO problem is typically NP-hard, leading to either computational inefficiency or approximate solutions with suboptimal approximation ratios. To remedy this problem, we

*Corresponding author: Xutong Liu. The work of Xiangxiang Dai was supported by the National Natural Science Foundation of China (625B2163). The work of John C.S. Lui was supported by RGC GRF-1420292.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on July 16,2026 at 05:50:42 UTC from IEEE Xplore. Restrictions apply.

propose the novel idea to bypass the above hard CO problem, and leverage the relaxation and rounding (RR) approach, which solves an easier **continuous** optimization problem that generally attains improved approximation guarantee and time complexity. A key challenge arises because RR, originally designed for *offline CO*, requires careful adaptation for *online settings*. *The flexibility in choosing the relaxation function F and rounding procedure σ complicates ensuring both low regret and low violation simultaneously.*

Third, to solve the above challenge, we discover general and unified properties for suitable (F, σ), which we call "**nice RR conditions**". These conditions not only build the key connection between the regret of C2MAB-V and the over-estimation terms, which can be bounded by CMAB-type analysis, but also enables us to bound the long-term violation based on our novel martingale construction techniques. Equipped with RR conditions, C2MAB-V is guaranteed to achieve $\tilde{O}(\sqrt{T})$ regret and violation with $\tilde{O}(1/\sqrt{T})$ diminishing rate. *In our analysis, we address several technical challenges to handle the non-linear reward function and the partial feedback, which are not considered by existing constraint-aware MAB works and are essential to cover different application scenarios later.*

Our final contribution lies in rigorously validating the utility of our framework, rather than merely prototyping it under the assumption that RR conditions hold. We undertake substantial efforts to demonstrate its applicability across a diverse range of applications, including mobile crowdsensing, network routing, and multi-LLM selection, accommodating versatile reward functions (e.g., linear, conjunctive, disjunctive, and submodular) and general feedback models (e.g., semi-bandit and cascading feedback). *For each application, we identify non-trivial relaxation functions and rounding procedures that satisfy RR conditions*, delivering *new and improved* regret bounds and/or computational efficiency guarantees. Furthermore, we evaluate the effectiveness of C2MAB-V by comparing it against benchmark algorithms across all three applications, using both synthetic and real-world datasets. Empirical results consistently show that C2MAB-V outperforms baselines, achieving lower regret while adhering to cost constraints.

To summarize, we make the following four contributions. (1) We propose a new constaint-aware CMAB problem to handle the cost of pulling arms and the long-term cost constraints. (2) We design the C2MAB-V framework that uses the relaxation and rounding technique to tackle the C2MAB-V problem with improved approximation guarantee and/or computational efficiency. (3) We discover the general RR conditions and under these conditions, C2MAB-V has sublinear regret, fast diminishing violation, and good computational efficiency. (4) We study concrete applications that satisfy the RR conditions with improved theoretical guarantees, and show superior performance of C2MAB-V algorithm in experiments using both synthetic and real-world datasets.

II. Related Work

Table I compares our C2MAB-V framework with related constraint-aware MAB works. To our knowledge, we are the first to address CMAB with general reward functions, long-term stochastic costs, and partial feedback, proposing a unified, computationally efficient solution via the Relaxation and Rounding (RR) approach with regret and violation guarantees.

MAB problem has been extensively studied in the literature (c.f. [17], [18], [1]). Different from [6], [7], [8] (rows 1-3 in Table I), we focus on *combinatorial MAB setting where multiple arms (combinatorial arms) are selected in each round, and the rewards are nonlinear combination of these arms, rather than the linear summation.* Stochastic CMAB has been studied extensively (rows 5–6), starting with linear CMAB [19], improved by [20], [21], [13]. Variants include cascading bandits [22], [23], influence maximization bandits [24], matroid bandits [25], CMAB for resource allocation [11], and CMAB with general reward functions [26], [4], [13]. Our work differs in that we further consider the costs of pulling arms and the long-term cost constraint requirements in choosing actions.

There is another line of works which consider resource consumption and budget constraint for stochastic MAB/CMAB, most notably, bandit with budgets [27], [28], [29] and bandit with knapsacks (BwK) [9], [14], [10]. For these works (row 4 in Table I), the optimal stopping time is considered since the learning process terminates when resources are depleted. Our problem differs, as *our long-term constraints allow arm selection to continue indefinitely*. Among BwK works, [14] is the closest to our work (row 7 in Table I), but is limited to linear CMAB with semi-bandit feedback. C2MAB-V generalizes this, degrading to their algorithm under linear relaxation F_1 and the dependent rounding procedure σ_1 (see Section VI-A). Another notable technical difference between [14] and our work is that we do not rely on the special negative correlation property of RR. Without this assumption, *our proof technique is more general and gives more flexibility on the choice of (F, σ) for a wide class of non-linear reward functions*.

Application-wise, [15] (row 8 in Table I) addresses web link selection with multiple plays, a special case of C2MAB-V with linear rewards and semi-bandit feedback. As a result, their technique is different and much simpler than ours. [30] studies the real-time traffic scheduling and task assignments in crowdsourcing with fairness constraints, but also adopts linear rewards and semi-bandit feedback. [16] (row 9 in Table I) explores submodular rewards with known costs, but extending to unknown stochastic costs, as in our work, is non-trivial.

RR is an effective approach for offline CO problems [31], [32], [33]. We summarize the existing results for offline RR methods in Table II. However, it is still *unclear and non-trivial how to use RR for the CMAB with non-linear rewards and partial feedback*. As far as we know, we are the first to integrate RR into non-linear CMAB, providing general conditions for regret and violation bounds.

III. Model and Preliminaries

In this paper, we use $[n]$ to denote $\{1, \ldots, n\}$, and **boldface letters** to represent random variables. We formulate the constraint-aware CMAB problem as follows. Consider a set $\mathcal{E} = [L]$ of L *base arms* (e.g., network links) and

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on July 16,2026 at 05:50:42 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Summary and comparison of the related constrained-aware MAB works.

Algorithm	Combinatorial Arms?	Non-linear Reward?	Unknown Stochastic Cost?	Cost Constraint	Partial Feedback?*
POND, [6]	×	×	✓	Long term	NA
OptPess-LP, [7]	×	×	✓	Long term	NA
Pessimistic-Optimistic, [8]	×	×	✓	Long term	NA
BwK† and its variants, [9], [10]	×	×	✓	BwK†	NA
CUCB-CRA, [11]	✓	✓	×	NA	×
CUCB-T and its variants, [4], [12], [13]	✓	✓	×	NA	✓
SemiBwK-RRS, [14]	✓	×	✓	BwK†	×
Constrained UCB, [15]	✓	×	×	Long term	×
AFSM-UCB, [16]	✓	✓	×	Per round‡	×
C2MAB-V, ours	✓	✓	✓	Long term	✓

† BwK means the bandit with knapsacks; * Partial feedback can handle partially observed arms; ‡ Per round means the action satisfies the cost constraint in each round.

TABLE II: Summary of the offline optimization algorithms and rounding procedures for RR methods.

Reward $F(x,w)$	Cost $G(x,w)$	Combinatorial Set $\mathcal{F}$	Oracle, Guarantee, Complexity	Rounding
Linear	Linear	Cardinality $\leq K$	LP [34], $\alpha_F = 1$, $O(L^{2.5})$ §	Dependent Rounding [32]
Linear	Linear	Partition	LP [34], $\alpha_F = 1$, $O(L^{2.5})$	KPR Rounding [35]
Linear	Linear	1-Matroid*	LP [34], $\alpha_F = 1$, $O(L^{2.5})$	Randomized Swap Rounding [33]
Linear	Submodular†	1-Matroid	Greedy [36], $\alpha_F = 1$, $\alpha_G = 1 - 1/e - \varepsilon$, $O(L^{4.5}/\varepsilon)$ ‡	Randomized Swap Rounding [33]
Monotone Submodular	Linear	1-Matroid	Continuous Greedy [36], $\alpha_F = 1 - 1/e$, $O(L^{4.5})$	Randomized Swap Rounding [33]
Monotone Submodular	Convex	1-Matroid	Submodular FW [37],$\alpha_F = 1 - 1/e - \varepsilon$, $O(L^{2.5}/\varepsilon)$	Randomized Swap Rounding [33]
Non-Monotone Submodular	Linear	1-Matroid	Shrunken FW [37], $\alpha_F = 1/e - \varepsilon$, $O(L^{2.5}/\varepsilon)$	Randomized Swap Rounding [33]

* 1-Matroid includes cardinality (uniform matroid), partition (partition matroid), spanning tree (graphic matroid), etc.; § α_F is the approximation ratio to the relaxed optimization problem in Eq. (4); † The constraint is in the form $G(x,w) \geq b$; ‡ α_G means solution x satisfies the constraint approximately, i.e., $G(x,w) \geq \alpha_G b$;

a non-empty set $\mathcal{F} \subseteq 2^{\mathcal{E}}$ of *feasible actions* (e.g., routing paths), constrained by combinatorial structures detailed in Section III-C. We call actions $S \in \mathcal{F}$ *feasible actions*. Each item in $\mathcal{E}$ is associated with a vector of stochastic weights (rewards) $\boldsymbol{w} \in \{0,1\}^L$, drawn from a distribution P_w with expected weights $w = \mathbb{E}_{\boldsymbol{w}\sim P_w}[\boldsymbol{w}]$. The reward for an action $S \in \mathcal{F}$ under weights $\boldsymbol{w}$ is $\hat{f}(S,\boldsymbol{w})$, with expected reward $\mathbb{E}_{\boldsymbol{w}\sim P_w}[\hat{f}(S,\boldsymbol{w})] = f(S,w)$, as in [4], [12], [13].[1] Each item in $\mathcal{E}$ also incurs stochastic costs $\boldsymbol{c} \in [0,1]^L$, drawn from a distribution P_c with expected costs $c = \mathbb{E}_{\boldsymbol{c}\sim P_c}[\boldsymbol{c}]$. The cost for action $S \in \mathcal{F}$ under costs $\boldsymbol{c}$ is $\hat{g}(S,\boldsymbol{c})$, with expected cost $\mathbb{E}_{\boldsymbol{c}\sim P_c}[\hat{g}(S,\boldsymbol{c})] = g(S,c)$. Remark that we do not assume random weights/costs are independent across arms but only require $f(S,w)$ and $g(S,c)$ depend only on S and the expected weights/costs, following [12], [4], [13]. The forms of reward and cost functions are detailed in Section III-A later on.

The agent interacts with the network environment over T rounds. In round t, the agent selects a feasible action $\boldsymbol{S}_t \in \mathcal{F}$ based on past feedback, receives reward $\hat{f}(\boldsymbol{S}_t,\boldsymbol{w}_t)$ and incurs cost $\hat{g}(\boldsymbol{S}_t,\boldsymbol{c}_t)$, where $\{\boldsymbol{w}_t,\boldsymbol{c}_t\}_{t=1}^T$ are i.i.d. from P_w and P_c.The goal is to devise a policy $\pi = \{\boldsymbol{S}_1,\ldots,\boldsymbol{S}_T\}$ that maximizes total expected reward while ensuring the long-term cost constraint $\sum_{t=1}^T \hat{g}(\boldsymbol{S}_t,\boldsymbol{c}_t)/T \leq b$. The optimal action S^* is defined as the maximizer of the following constrained optimization problem:

$$\arg\max_{S\in\mathcal{F}} f(S,w) \quad \text{s.t.} \quad g(S,c) \leq b. \tag{1}$$

Note that if the agent knew P_w and P_c, S^* would be the stationary action that maximizes the expected reward and ensures that the cost constraint is satisfied asymptotically, i.e.. $\lim_{T\to\infty}\sum_{t=1}^T \hat{g}(S^*,\boldsymbol{c}_t)/T \leq b$. This is **NP-hard** due to the cost constraint and complicated reward function (e.g., resembling a knapsack problem for linear f and submodular reward functions), unlike unconstrained CMAB that solves unconstrained problem $\max\{f(S,w) : S \in \mathcal{F}\}$ [23]. So the performance of the policy π is measured by α-*scaled regret* due to this computational hardness:

$$R_\pi^\alpha(T) = \mathbb{E}[\sum_{t=1}^T R^\alpha(\boldsymbol{S}_t,\boldsymbol{w}_t)] = \mathbb{E}[\sum_{t=1}^T \alpha f(S^*,w) - f(\boldsymbol{S}_t,w)], \tag{2}$$

where $R^\alpha(\boldsymbol{S}_t,\boldsymbol{w}_t) = \alpha\hat{f}(S^*,\boldsymbol{w}_t) - \hat{f}(\boldsymbol{S}_t,\boldsymbol{w}_t)$ is the instantaneous regret at time t. To measure how well the constraint is satisfied, we define the constraint *violation* of the policy π:

$$V_\pi(T) = \left[\frac{1}{T}\sum_{t=1}^T \hat{g}(\boldsymbol{S}_t,\boldsymbol{c}_t) - b\right]_+, [x]_+ = \max(x,0). \tag{3}$$

We say a policy π has violation with diminishing rate $\tilde{O}(T^{-\gamma})$ if $V_\pi(T) = \tilde{O}(T^{-\gamma})$, where $\gamma > 0$ describes how fast the violation diminishes when T increases. A good policy should take both regret and violation into consideration, achieving *sub-linear* regret as well as violation with *fast diminishing* rate.

A. Reward and Cost Functions

The reward function $f(S,w)$ satisfies *monotonicity* (non-decreasing in w) and B-*Lipschitz continuity* ($|f(S,w) - f(S,w')| \leq B\sum_{i\in S}|w_i - w_i'|$), as in [4], [13], [38]. Besides the general reward function, we will study four reward functions as special cases for three applications in section VI: (a) linear rewards $f(S,w) = \sum_{i\in S} w_i$ (e.g., throughput in routing), (b) conjunctive cascading $f(S,w) = \prod_{i\in S} w_i$, (c) disjunctive cascading $f(S,w) = 1 - \prod_{i\in S}(1-w_i)$, and (d) submodular rewards, where for all subsets $S,T \subseteq [L]$, $f_4(S,w) + f_4(T,w) \geq f_4(S\cup T,w) + f_4(S\cap T,w)$ and the above monotonicity and Lipschitz coninuity condition are satisfied. Careful readers may note that reward functions f_1, f_2, and f_3 are special cases of f_4. However, we design and analyze distinct algorithm components tailored to each, achieving better

[1] Unless $\hat{f}$ is linear in $\boldsymbol{w}$, $\hat{f}$ and f are different functions. Same for $\hat{g}$, g.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on July 16,2026 at 05:50:42 UTC from IEEE Xplore. Restrictions apply.

performance guarantees, rather than treating them solely as submodular functions. Costs are linear, $g(S,c)=\sum_{i\in S}c_i$.

B. Feedback Models

We consider two feedback models critical for network optimization, such as routing: *semi-bandit* and *cascading* feedback. In *semi-bandit feedback*, the agent observes the weights (rewards) of all selected items in action $\boldsymbol{S}_t$ at time t, i.e., $(\boldsymbol{w}_{t,\boldsymbol{S}_{t,1}},\dots,\boldsymbol{w}_{t,\boldsymbol{S}_{t,|\boldsymbol{S}_t|}})$, and all associated costs $(\boldsymbol{c}_{t,\boldsymbol{S}_{t,1}},\dots,\boldsymbol{c}_{t,\boldsymbol{S}_{t,|\boldsymbol{S}_t|}})$. In *cascading feedback*, the agent observes only the first $\boldsymbol{O}_t$ items of $\boldsymbol{S}_t$ based on a stopping criterion. For example, in routing, we set $\boldsymbol{O}_t=\min\{k:\boldsymbol{w}_{t,\boldsymbol{S}_{t,k}}=0\}$ or $\boldsymbol{O}_t=|\boldsymbol{S}_t|$, where $\boldsymbol{w}_{t,\boldsymbol{S}_{t,k}}=0$ indicates a link failure, halting observation (e.g., packet drop). The agent observes weights $\boldsymbol{w}_{t,\boldsymbol{S}_{t,k}}$ for $k\le\boldsymbol{O}_t$ and all costs $(\boldsymbol{c}_{t,\boldsymbol{S}_{t,1}},\dots,\boldsymbol{c}_{t,\boldsymbol{S}_{t,|\boldsymbol{S}_t|}})$. Semi-bandit feedback is a special case of cascading feedback when $\boldsymbol{O}_t=|\boldsymbol{S}_t|$. These models reflect partial observability in real-world network scenarios [39].

C. Long-term Constraints and Combinatorial Feasible Sets

The long-term cost constraint, defined in Eq. (3), requires the average cost $\sum_{t=1}^{T}\hat{g}(\boldsymbol{S}_t,\boldsymbol{c}_t)/T\le b$, widely used in online optimization and bandit problems [40], [10], [41]. Unlike per-round constraints, this allows costs to exceed b during exploration, but ensures long-term compliance, crucial for energy-efficient routing or resource-constrained load balancing [10], [5]. The feasible set $\mathcal{F}$ comprises *linearizable* combinatorial structures, enabling efficient randomized rounding [14]. Formally, $\mathcal{F}$ is defined by a convex polytope $\mathcal{P}(\mathcal{F})=\text{conv}\{\mathbb{I}_S:S\in\mathcal{F}\}$, where $\mathbb{I}_S=\{Y_1,...,Y_L\}\in\{0,1\}^L$ is the indicating vector such that the i-th element $Y_i=1$ if and only if $i\in S$. Matroid constraints, such as cardinality (e.g., at most K links), partition matroids (e.g., channel assignments), and spanning trees (e.g., routing trees), are key examples.

IV. Algorithm Design

A. Motivation for Relaxation and Rounding (RR)

Generally speaking, to achieve an α-scaled regret, `C2MAB-V` uses UCB/LCB estimates for weights w and costs c in the constrained optimization problem Eq. (1) to select an α-approximate action S_t per round. The cost constraint $g(S,c)\le b$ makes solving Eq. (1) computationally challenging, often NP-hard [4]. Prior works [26], [16] rely on offline oracles with suboptimal approximation ratios α_f or high complexity. We propose an RR approach, solving a relaxed *continuous* optimization problem Eq. (4) with approximation ratio $\alpha_F\ge\alpha_f$, followed by rounding to a discrete action. This reduces complexity and improves regret by $(\alpha_F-\alpha_f)T$. For example, in *mobile crowdsensing* (Section VI-A), Eq. (1) resembles a 0-1 knapsack problem for selecting links, while the relaxed problem is a linear program solvable in $O(L^{2.5})$ time with $\alpha_F=1$ [34]. In contrast, the best discrete algorithm achieves $\alpha_f=1-\epsilon$ in $O(L^{1.5}/\epsilon)$ time [42], yielding an additional T/L regret for $\epsilon=1/L$. RR thus enhances efficiency and performance in network optimization.

B. RR Approach for Constrained Optimization

The RR approach for solving the offline constrained optimization problem involves two steps:

Step 1: Solve the relaxed problem:

$$\max\{F(x,w):x\in\mathcal{P}(\mathcal{F}),G(x,c)\le b\},\tag{4}$$

where F is the relaxation of reward function f, $G(x,c)=\sum_{i\in[L]}c_ix_i$ is the linear relaxation of cost function g, and $\mathcal{P}(\mathcal{F})$ is the polytope induced by the combinatorial feasible set $\mathcal{F}$ (e.g., matroids for routing trees).

Step 2: Round the continuous solution x to a discrete action $S=\sigma(x)\in\mathcal{F}$ using a rounding procedure σ.

Though RR has been widely used for offline CO problems, it is not clear how to guarantee that RR can produce randomized *online solutions* which not only explore the solution space efficiently but also yield large rewards and fast diminishing violation. Our observation is that with different relaxation F and rounding procedure σ, the solution quality and the computational efficiency may vary. Therefore, we ask the following core question:

"*What general conditions do we need for F and σ to guarantee our learning policy has low regret, fast diminishing violation, and good computational efficiency?*"

To answer this question, we come up with the following conditions, which we call the "nice" RR conditions.

Definition 1 (Nice RR Conditions)**.** *(F,σ) forms a nice RR pair with approximation ratio α if: (1) An α-approximate solution to Eq. (4) is polynomial-time solvable; (2) σ generates $S\sim\sigma(x)\in\mathcal{F}$ in polynomial time, with $\mathbb{E}[\mathbb{I}_S]=x$, where the indicating vector $\mathbb{I}_S=\{Y_1,...,Y_L\}\in\{0,1\}^L$ with $Y_i=1$ if and only if $i\in S$; (3) $f(S,w)=F(\mathbb{I}_S,w)$ for any feasible action $S\in\mathcal{F}$; (4) (F,σ) is convex preserving* [2] *, i.e., $\mathbb{E}_{S\sim\sigma(x)}[F(\mathbb{I}_S,w)]\ge F(x,w)$.*

Remark 1. The general *nice RR conditions* ensure three good properties: sublinear regret, fast-diminishing violation, and polynomial-time complexity. Specifically: (1) Polynomial-time solvability of the relaxed constrained optimization Eq. (4) ensures efficient action selection per round. (2) The rounding procedure σ produces feasible actions $S\in\mathcal{F}$ with $\mathbb{E}[\mathbb{I}_S]=x$, satisfying cost constraints in expectation when $G(x,c)<b$. Together, Conditions (1) and (2) bypass the NP-hard discrete problem Eq. (1). (3) $f(S,w)=F(\mathbb{I}_S,w)$ ensures the discrete reward matches the relaxed reward. (4) $\mathbb{E}_{S\sim\sigma(x)}[F(\mathbb{I}_S,w)]\ge F(x,w)$ guarantees the expected reward quality is at least as good as the relaxed solution. Note that one can also check that relaxation function F and rounding procedure σ of Table II satisfies our general RR condition. Later in Section VI, we will design nice RR pairs (F,σ) for reward functions considered in Section III-A, with the above four conditions met.

C. C2MAB-V for Online Learning

`C2MAB-V` (Algorithm 1) that integrates the upper/lower confidence bound framework and the RR approach is parameterized

[2]The name is motivated by the similarity of the Jensen's inequality. for convex functions with $\mathbb{E}_{S\sim\sigma(x)}[F(\mathbb{I}_S,w)]\ge F(\mathbb{E}_{S\sim\sigma(x)}[\mathbb{I}_S],w)$.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on July 16,2026 at 05:50:42 UTC from IEEE Xplore. Restrictions apply.

by the number of base arms L, the allowed failure probability δ, combinatorial feasible set $\mathcal{F}$, the relaxation function for rewards and costs F, G and the rounding procedure σ. For each arm i, $\boldsymbol{T}_{w,t,i}$ and $\boldsymbol{T}_{c,t,i}$ track observed weights and costs, where $\boldsymbol{T}_{w,t,i} \neq \boldsymbol{T}_{c,t,i}$ in general. Let $\hat{\boldsymbol{w}}_{t,i}, \hat{\boldsymbol{c}}_{t,i}$ denote the empirical mean for the true reward w_i and true cost c_i. After initialization (line 2), in each round t, we compute the confidence radius $\rho_{w,t,i}$ and $\rho_{c,t,i}$ (line 6), which control the level of exploration and are larger for less-explored arms. In line 7, we use the confidence radius to obtain the "optimistic" estimates of w and c, which we denote as the UCB $\bar{\boldsymbol{w}}_{t,i}$ and LCB $\underline{\boldsymbol{c}}_{t,i}$. Line 10 solves the constrained problem of Eq. (1) with UCB/LCB values and outputs a continuous solution $\boldsymbol{x}_t$. Line 12 rounds the $\boldsymbol{x}_t$ into a feasible solution S_t. Finally, we update the statistics using the observed weights and costs, depending on the feedback models discussed in Section III-B.

Algorithm 1 C2MAB-V: constraint-aware CMAB with versatile reward functions

1: **Input:** size of base arms L, confidence parameter $\delta \in (0,1]$, cost constraint b, combinatorial feasible set $\mathcal{F}$, relaxed reward function F, relaxed cost function G, rounding procedure σ, parameters $\alpha_w, \alpha_c \in (0,1]$.
2: **Initialization:** For each arm $i \in [L]$, $\hat{\boldsymbol{w}}_{1,i} \leftarrow \boldsymbol{w}_{0,i}$, $\boldsymbol{T}_{w,1,i} \leftarrow 1$, $\hat{\boldsymbol{c}}_{1,i} \leftarrow \boldsymbol{c}_{0,i}, \boldsymbol{T}_{c,1,i} \leftarrow 1$.
3: **for** $t = 1, 2, ..., T$ **do**
4: // Compute UCBs and LCBs.
5: **for** $i \in [L]$ **do**
6: $\rho_{w,t,i} \leftarrow \sqrt{\ln(\frac{2\pi^2 K t^3}{3\delta})/2\boldsymbol{T}_{w,t,i}}$, $\rho_{c,t,i} \leftarrow \sqrt{\ln(\frac{2\pi^2 K t^3}{3\delta})/2\boldsymbol{T}_{c,t,i}}$.
7: $\bar{\boldsymbol{w}}_{t,i} \leftarrow \min\{\hat{\boldsymbol{w}}_{t,i} + \alpha_w \rho_{w,t,i}, 1\}$, $\underline{\boldsymbol{c}}_{t,i} \leftarrow \max\{\hat{\boldsymbol{c}}_{t,i} - \alpha_w \rho_{c,t,i}, 0\}$.
8: **end for**
9: // Solve the constrained optimization problem.
10: $\boldsymbol{x}_t \leftarrow \arg\max\{F(x, \bar{\boldsymbol{w}}_t) : x \in \mathcal{P}(\mathcal{F}), G(x, \underline{\boldsymbol{c}}_t) \leq b\}$.
11: // Obtain $\boldsymbol{S}_t$ via randomized rounding procedure σ.
12: Obtain $\boldsymbol{S}_t \sim \sigma(\boldsymbol{x}_t)$.
13: Play $\boldsymbol{S}_t$, observe $\boldsymbol{O}_{w,t}, \boldsymbol{w}_{\boldsymbol{S}_{t,k}}$ for $k \leq \boldsymbol{O}_{\boldsymbol{w},t}$ and observe $\boldsymbol{O}_{c,t}, \boldsymbol{c}_{t,\boldsymbol{S}_{t,l}}$ for $l \leq \boldsymbol{O}_{c,t}$.
14: // Update statistics using observed weights and costs.
15: **for** $k = 1, ..., \boldsymbol{O}_{w,t}$, $l = 1, ..., \boldsymbol{O}_{c,t}$ **do**
16: $i \leftarrow \boldsymbol{S}_{t,k}$, $j \leftarrow \boldsymbol{S}_{t,l}$.
17: $\boldsymbol{T}_{w,t+1,i} \leftarrow \boldsymbol{T}_{w,t,i} + 1$, $\boldsymbol{T}_{c,t+1,j} \leftarrow \boldsymbol{T}_{c,t,j} + 1$.
18: $\hat{\boldsymbol{w}}_{t+1,i} \leftarrow \frac{\boldsymbol{T}_{w,t,i}\hat{\boldsymbol{w}}_{t,i} + \boldsymbol{w}_{t,i}}{\boldsymbol{T}_{w,t+1,i}}$, $\hat{\boldsymbol{c}}_{t+1,j} \leftarrow \frac{\boldsymbol{T}_{c,t,j}\hat{\boldsymbol{c}}_{t,j} + \boldsymbol{c}_{t,i}}{\boldsymbol{T}_{c,t+1,j}}$.
19: **end for**
20: **end for**

V. Theoretical Analysis

In this section, we analyze the regret and violation of C2MAB-V. The algorithm runs in polynomial time, as it operates over T rounds, with the most time-consuming step being the RR procedure, which is polynomial-time due to the conditions in Definition 1. We begin with a lemma stating two high-probability events: the empirical means of rewards and costs are close to their true means.

Lemma 1. *Let event $\mathcal{N}_w$ be the event that, for every round t and arm i, $|\hat{\boldsymbol{w}}_{t,i} - w_i| < \rho_{w,t,i} = \sqrt{\ln(\frac{2\pi^2 K t^3}{3\delta})/2\boldsymbol{T}_{w,t,i}}$ and let event $\mathcal{N}_c$ be the event that, for every round t and arm i, $|\hat{\boldsymbol{c}}_{t,i} - c_i| < \rho_{c,t,i} = \sqrt{\ln(\frac{2\pi^2 K t^3}{3\delta})/2\boldsymbol{T}_{c,t,i}}$ then $\Pr\{\mathcal{N}_w\} \geq 1 - \delta/2, \Pr\{\mathcal{N}_c\} \geq 1 - \delta/2$.*

A. Regret Analysis

For the regret analysis, our idea is to decompose the total regret into two parts: the RR regret and the over-estimation regret. For the RR regret, we leverage on the freedom of choosing (F, σ) pair in RR and identify fundamental properties (that nice RR pairs (F, σ) should satisfy) so that the RR regret can be bounded. Thanks to our decomposition and the RR condition, the total regret now reduces to the over-estimation regret, which can be bounded by discovering the connection to the key counterparts of CMAB proof techniques.

To state the regret bound, we need to give some definitions. Let $p_{t,S}$ be the probability of *full observations* for S, i.e., the probability that all base arms in S are observed. Let $p^* = \min_{t\in[T], S\in\mathcal{F}} p_{t,S}$ be the minimum probability of full observation for any feasible action in any round and $f_{\max} = \max_{S\in\mathcal{F}} f(S, w) \leq BK$ be the maximum reward any feasible solution could achieve, where $K = \max_{S\in\mathcal{F}} |S|$ is the maximum number of base arms that constitutes a feasible action. One of our main results is the following regret bound.

Theorem 1. *For any $\delta \in (0,1)$ and any reward function $f(S, w)$ that satisfies monotonicity and B-Lipschitz continuity condition, if the relaxed function and rounding procedure (F, σ) forms a nice RR pair with approximation ratio α (see Definition 1), then the α-approximate regret of C2MAB-V is bounded as follows:*

$$R^{\alpha}_{ALG}(T) \leq \frac{2B}{p^*}\sqrt{2KLT\ln(\frac{2\pi^2 LT^3}{3\delta})} + f_{\max}(L + \delta T). \quad (5)$$

Proof (sketch). Let S^* be the optimal solution for Eq. (1) with the true parameters w and c. Let $\boldsymbol{x}_t$ be the α-approximate solution of $\max\{F(x, \bar{\boldsymbol{w}}_t) : x \in \mathcal{P}(\mathcal{F}), G(x, \underline{\boldsymbol{c}}_t) \leq b\}$ and $\boldsymbol{S}_t$ be the feasible solution given by $\boldsymbol{S}_t = \sigma(\boldsymbol{x}_t)$ as in line 12. When the high probability events $\mathcal{N}_w$, $\mathcal{N}_c$ happen, we have $w_i \leq \hat{\boldsymbol{w}}_{t,i} + \rho_{w,t,i} = \bar{\boldsymbol{w}}_{t,i}$ and $c_i \geq \hat{\boldsymbol{c}}_{t,i} - \rho_{c,t,i} = \underline{\boldsymbol{c}}_{t,i}$. We first apply the monotonicity property and the nice RR pair conditions to reduce the expected regret at t to the over-estimation regret,

$$\begin{aligned}
&\mathbb{E}[R^{\alpha}(\boldsymbol{S}_t, \boldsymbol{w}_t)] = \mathbb{E}[\alpha f(S^*, w) - f(\boldsymbol{S}_t, w)] && (6)\\
&\leq \mathbb{E}[\alpha f(S^*, \bar{\boldsymbol{w}}_t) - f(\boldsymbol{S}_t, w)] && (7)\\
&\leq \mathbb{E}[F(\boldsymbol{x}_t, \bar{\boldsymbol{w}}_t) - f(\boldsymbol{S}_t, w)] && (8)\\
&= \mathbb{E}[\underbrace{(F(\boldsymbol{x}_t, \bar{\boldsymbol{w}}_t) - \mathbb{E}_{\boldsymbol{S}_t\sim\sigma(\boldsymbol{x}_t)}[F(\mathbb{I}_{\boldsymbol{S}_t}, \bar{\boldsymbol{w}}_t)])}_{\text{RR regret}} \\
&\quad + \underbrace{(f(\boldsymbol{S}_t, \bar{\boldsymbol{w}}_t) - f(\boldsymbol{S}_t, w))}_{\text{Over-estimation regret}}] && (9)\\
&\leq \mathbb{E}[\underbrace{f(\boldsymbol{S}_t, \bar{\boldsymbol{w}}_t) - f(\boldsymbol{S}_t, w)}_{\text{Over-estimation regret}}], && (10)
\end{aligned}$$

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on July 16,2026 at 05:50:42 UTC from IEEE Xplore. Restrictions apply.

where Eq. (6) is because the regret definition, Eq. (7) is due to monotonicity of f, Eq. (8) is due to $f(S^*, \bar{\boldsymbol{w}}_t) = F(\mathbb{I}_{S^*}, \bar{\boldsymbol{w}}_t)$ by condition (3) in Definition 1 and $\mathbb{I}_{S^*}$ is one feasible solution of optimization problem $\max\{F(x, \bar{\boldsymbol{w}}_t) : x \in \mathcal{P}(\mathcal{F}), G(x, \underline{\boldsymbol{c}}_t) \leq b\}$, Eq. (9) is because $\mathbb{E}[\mathbb{E}_{\boldsymbol{S}_t \sim \sigma(\boldsymbol{x}_t)}[F(\mathbb{I}_{\boldsymbol{S}_t}, \bar{\boldsymbol{w}}_t)]] = \mathbb{E}[F(\mathbb{I}_{\boldsymbol{S}_t}, \bar{\boldsymbol{w}}_t)] = \mathbb{E}[f(\boldsymbol{S}_t, \bar{\boldsymbol{w}}_t)]$, Eq. (10) holds due to condition (4) in definition 1. Here we can see similar counterparts in [20], [23], [13] that bounds the over-estimation regret in Eq. (10) (but with expectation taken over the additional randomness from the rounding procedure for $\boldsymbol{S}_t$), where the main challenge is to handle the non-linear reward, the partial observation and the additional randomness. With usage of the B-Lipschitz condition of f and the careful analysis over observation probability, we bound Eq. (10) by the desired terms in Eq. (5). □

B. Violation Analysis

For the violation analysis, one way is to follow the [14] by additionally assuming the stronger negative correlation property of RR. However, we find ways to rely on weaker assumptions (only with condition (2) in Definition 1, which is also assumed in [14]) and bound the violation by using martingale techniques. Without the negative correlation assumption, our proof technique is much more general and gives more flexibility on the choice of (F, σ) for different reward functions. Now we give the following high probability bound for the violation.

Theorem 2. *The violation of* C2MAB-V *(Algorithm 1) is bounded as follows with probability at least* $1 - \delta$*:*

$$V_{ALG}(T) \leq 2\sqrt{\tfrac{2KL}{T}\ln(\tfrac{2\pi^2 LT^3}{3\delta})} + KL/T. \tag{11}$$

Proof (sketch). To prove the violation bound, we consider the violation when event $\mathcal{N}_c$ happens, with probability at least $1 - \delta/2$ (by Lemma 1). Under $\mathcal{N}_c$, we have $c_i \leq \bar{\boldsymbol{c}}_t \leq c_i + 2\rho_{c,t,i}$. From line 10 in Algorithm 1, we know that $b \geq G(\boldsymbol{x}_t, \underline{\boldsymbol{c}}_t) = \sum_{i\in[L]} \underline{\boldsymbol{c}}_{t,i}\boldsymbol{x}_{t,i} = \underline{\boldsymbol{c}}_t \cdot \boldsymbol{x}_t$ and then $V_{ALG}(T) = \left[\sum_{t=1}^{T}, \hat{g}(\boldsymbol{S}_t, \boldsymbol{c}_t)/T - b\right]_+$ is bounded by

$$V_{ALG}(T) \leq \frac{1}{T}(\underbrace{|\sum_{t=1}^{T}(\sum_{i\in\boldsymbol{S}_t}\boldsymbol{c}_{t,i} - \sum_{i\in\boldsymbol{S}_t} c_i)|}_{\text{term (a)}}$$
$$+ \underbrace{|\sum_{t=1}^{T}(\sum_{i\in\boldsymbol{S}_t} c_i - \sum_{i\in\boldsymbol{S}_t}\underline{\boldsymbol{c}}_{t,i})|}_{\text{term (b)}} + \underbrace{|\sum_{t=1}^{T}(\sum_{i\in\boldsymbol{S}_t}\underline{\boldsymbol{c}}_{t,i} - \underline{\boldsymbol{c}}_t\cdot\boldsymbol{x}_t)|}_{\text{term (c)}}). \tag{12}$$

Since we do not take expectation over violation, we have to construct martingale sequences and use Azuma-Hoeffding inequality to bound term (a) by $(\sqrt{2K^2T\ln(8/\delta)})$ with at least $1 - \delta/4$ probability. Similarly, term (c) is bounded by $(\sqrt{2K^2T\ln(8/\delta)})$ with at least $1-\delta/4$. Under event $\mathcal{N}_c$, term (b) is bounded by $2\sqrt{2KLT\ln(\frac{2\pi^2LT^3}{3\delta})}$. Using the union bounds, we get the desired results in Eq. (11). □

C. Discussions

The C2MAB-V framework generalizes existing CMAB models without long-term cost constraints [20], [23], [4], unifying general reward functions and feedback models. Unlike prior works designing specific conditions per reward or feedback, we propose universal *nice RR conditions* (Definition 1).

For regret with failure probability $\delta = 1/T$, C2MAB-V matches the performance of linear CMAB with semi-bandit feedback [20] when setting $B = 1$, $p^* = 1$, achieving a bound of the same order, close to the lower bound $\Omega(\sqrt{KLT})$ up to $\sqrt{\ln T}$. Compared with CMAB with cascading feedback [22], [23], our regret bound also matches with their results. The reason that our regret bound is tighter compared with other CMAB works in general is as follows. The major part of other CMAB works' regret is due to the over-estimation of unknown parameters, while we have an RR procedure part that causes additional regrets. However, the additional regret for the RR procedure is non-positive in expectation if Definition 1 is satisfied. *As for the p^* term, it is a coefficient that also appears in [22], [4], [12] for partial observation [22], [12] or the triggering probability of observing the base arms [4], which could be improved for specific problem instances (see Section VI-B for example), or removed by assuming Triggering Probability Modulated Lipschitz continuity for f as in [13].*

For the violation bound with $\delta = 1/T$, the violation diminishes at $O(\sqrt{KL\ln T/T})$. As $T \to \infty$, $V_\pi(T) \to 0$, achieving asymptotic zero violation. The cumulative violation is $O(\sqrt{KLT\ln T})$, matching the regret order. Henceforth, for some application scenarios (where the induced polytope $\mathcal{F}$ is down-closed and the relaxed reward function $F(x, w)$ is up-concave for x), trade-offs can be made between the regrets and the violation, so that the violation is 0 for *finite* time T with high probability. The trick is to tighten the constraint $g(S, c) \leq b$ by $g(S, c) \leq b(1 - \epsilon)$ but pay more regrets. With proper ϵ, we can prove $V_\pi(T) = 0$ while keeping the order of the regret unchanged. To get a violation bound for general (e.g., non-linear) cost function $g(S, c)$, one can equip $g(S, c)$ with the similar monotonicity, B'-Lipschitz continuity property and assume the relaxation $G(x, w)$ is concave-preserving in Definition 1. This generalization will be left for future work.

VI. Applications for C2MAB-V

In this section, we consider three representative applications that can fit into our framework. For each application, we present the RR pair (F, σ) that satisfies conditions in Definition 1 and show the regret bound, the violation bound, and the time complexity. We also provide a summary of the new results for different applications in Table III compared to existing works.

A. Mobile Crowdsensing

Mobile crowdsensing involves a platform where the agent aims to select K tasks from a total of L available tasks to assign to mobile device users (e.g., smartphone users) over T consecutive rounds [3]. Executing each task consumes a random amount of resources (e.g., data bandwidth), drawn from an unknown distribution. For task i, let $w_i \in [0, 1]$ be

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on July 16,2026 at 05:50:42 UTC from IEEE Xplore. Restrictions apply.

TABLE III: Summary of applications with 1-matroid combinatorial feasible set under versatile reward functions.

Reward Function	Offline Oracle	Approximation	α-Regret	Time Complexity
Linear	0-1 Knapsack [42]	$\alpha = 1 - 1/L$	$\tilde{O}(\sqrt{KLT})$	$O(L^{2.5}T)$
Linear	RR (Section VI-A)	$\alpha = 1$	$\tilde{O}(\sqrt{KLT})$	$O(L^{2.5}T)$
Conjunctive Cascade	0-1 Knapsack [42]	$\alpha = 1 - 1/L$	$\tilde{O}(\sqrt{KLT}/f^*)$	$O(L^{2.5}T)$
Conjunctive Cascade	RR (Section VI-B)	$\alpha = 1$	$\tilde{O}(\sqrt{KLT}/f^*)$	$O(L^{2.5}T)$
Disjunctive Cascade	Submodular Max [16]	$\alpha = \frac{1}{4(1+\varepsilon)}$	$\tilde{O}(\sqrt{KLT}/p^*)$	$O(\frac{KLT}{\ln(1+\varepsilon)})$
Disjunctive Cascade	RR (Section VI-C)	$\alpha = 1 - 1/e$	$\tilde{O}(\sqrt{KLT}/p^*)$	$O(L^{4.5}T)$

the unknown probability that task i is completed by the user and yields useful data, and let $c_i \in [0,1]$ be the expected resource consumption. The long-term constraint is that the average resource consumption over T rounds must remain below a threshold b. Our goal is to select K tasks to maximize the total amount of useful data collected, subject to the resource consumption constraint. For feedback, we adopt the semi-bandit feedback model for rewards and costs, which are revealed for all selected tasks.

In this application, the reward function $f_1(S,w) = \sum_{i\in S} w_i$ is the linear reward function and the feasible actions are $\mathcal{F}_1 = \{S \subseteq [L] : |S| = K\}$. We choose the relaxed reward function to be $F_1(x,w) = \sum_{i\in[L]} w_i x_i$ and the random rounding procedure σ_1 to be the dependent rounding procedure from [32]. Then the relaxed constraint optimization problem is $\max\{\sum_{i\in[L]} w_i x_i : \sum_{i\in[L]} x_i = K, 0 \le x_i \le 1\, \forall i, \sum_{i\in[L]} c_i x_i \le b\}$. Setting $\delta = 1/T$, we have the following corollary for our C2MAB-V algorithm to solve the mobile crowdsensing problem.

Corollary 1. *(F_1, σ_1) forms a nice RR pair with approximation ratio $\alpha = 1$, the regret is bounded by $2\sqrt{2KLT\ln(2\pi^2 LT^4/3)} + K(L+1)$, the violation is bounded by $2\sqrt{2KL\ln(2\pi^2 LT^4/3)/T} + KL/T$, with high probability (w.h.p.). The time complexity is $O(L^{2.5}T)$.*

B. Network Routing

In network routing [23], the agent aims to select a routing path to maximize the probability that all intermediate routers in the chosen path are operational. Consider a network with L available routers, each with a fixed but unknown uptime probability $w_i \in [0,1]$, influenced by varying network traffic and communication capacities. When the agent selects a router, the network system imposes a random cost with an unknown expected value $c_i \in [0,1]$ to account for the traffic load incurred. The agent observes whether the packet is successfully delivered through all routers and, if not, identifies the first router in the path that is down. The random costs for all selected routers are also observed. The objective is to choose K routers from the L available to maximize the packet delivery probability, while ensuring that the average cost over T rounds remains below a threshold b.

In this context, the reward function $f_2(S,w) = \prod_{i\in S} w_i$ is the conjunctive reward function, with feasible actions given by $\mathcal{F}_2 = S \subseteq [L] : |S| = K$. We adopt a relaxed function $F_2(x,w) = \prod_{i\in[L]} {w_i}^{x_i}$ and use the dependent rounding procedure σ_2 [32]. Then the relaxed constraint optimization problem is $\max\{\prod_{i\in[L]} {w_i}^{x_i} : \sum_{i\in[L]} x_i = K, 0 \le x_i \le 1\, \forall i, \sum_{i\in[L]} c_i x_i \le b\}$. The optimal solution S^* is equivalent to solving the linear program $\arg\max\{\sum_{i\in[L]} x_i \ln w_i : \sum_{i\in[L]} x_i = K, 0 \le x_i \le 1\, \forall i, \sum_{i\in[L]} c_i x_i \le b\}$ by taking $\ln F_2(x,w)$ as our objective. The reward feedback follows a cascading model: $\boldsymbol{O}_{w,t} = K$ if all routers are operational, or $\boldsymbol{O}_{w,t} = \min i \in [K] : \boldsymbol{w}_{t,\boldsymbol{s}_{t,i}} = 0$ otherwise. Let $f^* = f_2(S^*, w)$. Setting $\delta = 1/T$, we obtain:

Corollary 2. *(F_2, σ_2) forms a nice RR pair with approximation ratio $\alpha = 1$, the regret is bounded by $\frac{8}{f^*}\sqrt{2KLT\ln(2\pi^2 LT^4/3)} + L + 1$, the violation is bounded by $2\sqrt{2KL\ln(2\pi^2 LT^4/3)/T} + KL/T$, w.h.p. The time complexity is $O(L^{2.5}T)$.*

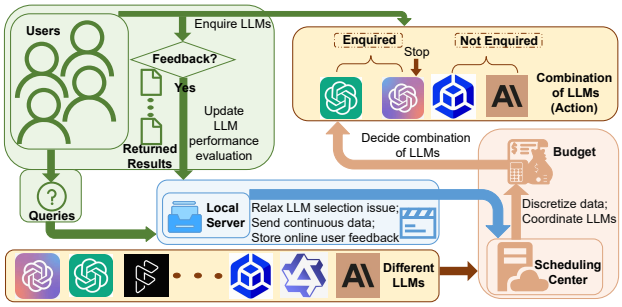


Fig. 1: Multi-LLM selection workflow via online RR: A cloud-based scheduler coordinates multiple LLMs, selecting them based on relaxed continuous data from a local server. The local server collects user feedback to improve online LLM evaluations.

C. Multi-LLM Selection

The growing availability of Large Language Model (LLM) APIs with heterogeneous performance and costs offers a unique opportunity for data-adaptive LLM selection [43], e.g., Investlm [44] is specifically designed for the financial sector. This motivates an "online" approach to optimize the multi-LLM selection problem, leveraging continuous user feedback to adapt to varying model performance and diverse application requirements. LLM cascade bandits [45], [46] enable real-time deployment of multiple LLM APIs, ensuring satisfactory outcomes through user interactions. This approach is critical for LLM-powered systems, such as chatbots [47], agent networks [48], and other interactive AI applications. As shown in Fig. 1, the multi-LLM selection problem is modeled as a sequential decision-making process over T rounds. At each

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on July 16,2026 at 05:50:42 UTC from IEEE Xplore. Restrictions apply.

round t, a user submits a query, and the learner selects an ordered list S_t of K LLM APIs (i.e., a super arm) from L candidate LLMs (i.e., base arms). Each LLM i has an unknown success probability $w_i \in [0,1]$ for user satisfaction. Following a statistically-based cost model [49], [50], for a query q from set $\mathcal{Q}$, LLM k uses deterministic input tokens $l_k^{\text{in}}(q)$ and random output tokens $l_k^{\text{out}}(q) \sim D^{\text{out}}k(q)$ [51], [52]. Given a query distribution D_q, the cost for LLM i is $y_{t,i} = (l_i^{\text{in}}(q_t) + l^{\text{out}}i(q_t))C_i$, where C_i is the cost per token. The goal is to estimate the unknown expected cost $c_i = \mathbb{E}[y_{t,i}]$. Users query LLMs sequentially from list S, stopping at the first satisfactory response or exhausting the list. The agent earns a reward of 1 if a response satisfies the user, otherwise 0. The objective is to recommend up to K LLM APIs to maximize the probability of user satisfaction while keeping the average cost over T rounds below threshold b.

In this application, the reward function $f_3(S,w) = 1 - \prod_{i\in S}(1-w_i)$ is the disjunctive reward function, with feasible actions $\mathcal{F}_3 = \{S \subseteq [L] : |S| \leq K\}$. Different to the conjunctive function $f_2(S,w)$ in Section VI-B, we can not simply use $1 - \prod_{i\in[L]} {w_i}^{x_i}$ as our relaxed function F_3, which is a concave function with respect to x. Our solution is to treat $f_3(S,w)$ as a *submodular* function and use its multi-linear extension $F_3(x,w) = \sum_{S\subseteq[L]} f_3(S,w) \prod_{i\in S} x_i \prod_{i\notin S}(1-x_i)$, which is convex-preserving along any directions $\mathbb{I}_{\{i\}} - \mathbb{I}_{\{j\}}$ for $i \neq j \in [L]$ [36]. We calculate the closed form $F_3(x,w) = 1 - \prod_{i\in[L]}(1-w_i x_i)$ of the above multi-linear extension and the relaxed constrained optimization problem is identified as $\max\{1 - \prod_{i\in[L]}(1-w_i x_i) : \sum_{i\in[L]} x_i \leq K, 0 \leq x_i \leq 1 \forall i, \sum_{i\in[L]} c_i x_i \leq b\}$, which can be efficiently solved by the continuous greedy algorithm [36] approximately with ratio $\alpha = (1-1/e)$ in $O(L^2)$ iterations. This is the best we can do unless P=NP. To ensure that condition (4) in Definition 1 holds, we also need to use a new rounding procedure called randomized swap rounding [33], which rounds x toward the convex-preserving directions. The cascading feedback is $\boldsymbol{O}_{w,t} = K$ if no LLM satisfies the user, or $\boldsymbol{O}_{w,t} = \min i \in [K] : \boldsymbol{w}_{t,\boldsymbol{S}_{t,i}} = 1$ otherwise. With $\delta = 1/T$, the `C2MAB-V` algorithm yields:

Corollary 3. *(F_3, σ_3) forms a nice RR pair with $\alpha = 1-1/e$, the regret is bounded by $\frac{2}{p^*}\sqrt{2KLT\ln(2\pi^2 LT^4/3)}+L+1$, the violation is bounded by $2\sqrt{2KL\ln(2\pi^2 LT^4/3)/T} + KL/T$, w.h.p. The time complexity is $O(L^{4.5}T)$.*

VII. Experiments

In this section, we provide the empirical results for the three aforementioned applications. Experiments were conducted on a device with an Intel Core i5-13600KF CPU @ 3.50GHz and 32 GB of memory. We used Gurobi 11.0.1 [53], a mathematical optimization solver, to address relaxed constraint optimization problems. Results are averaged over 10 seeds, reported with a 95% confidence interval.

Performance Metrics. Besides reward and cost, we evaluate performance using the reward-to-violation ratio, defined as the average per-round reward divided by the average per-round violation, i.e., $\frac{\sum_{\tau=1}^{t} f(\boldsymbol{S}_\tau, w)/t}{\sum_{\tau=1}^{t}[\hat{g}(\boldsymbol{S}_\tau, \boldsymbol{c}_\tau)/t - b]_+}$, where higher ratios indicate better performance.

Comparison Benchmarks. Comparisons include `CUCB`, [13], [54], which ignores constraints, `Thompson Sampling`, a mainstream probabilistic Bayesian online learning approach, [1] and ϵ-`Greedy` [55], which alternates between using empirical means and selecting uniformly based on the adaptive $\epsilon_t = \min\{1, \frac{2\sqrt{L}}{\sqrt{t}}\}$ (Here, works of [14], [16] are not compared due to being restricted to fixed costs or linear reward). The robustness of `C2MAB-V` on ablation study is validated by varying the parameters α_w, α_c with values of (0.3, 0.05), (1, 0.05), (0.3, 0.01), and (1, 0.01), referred to as (a), (b), (c), (d).

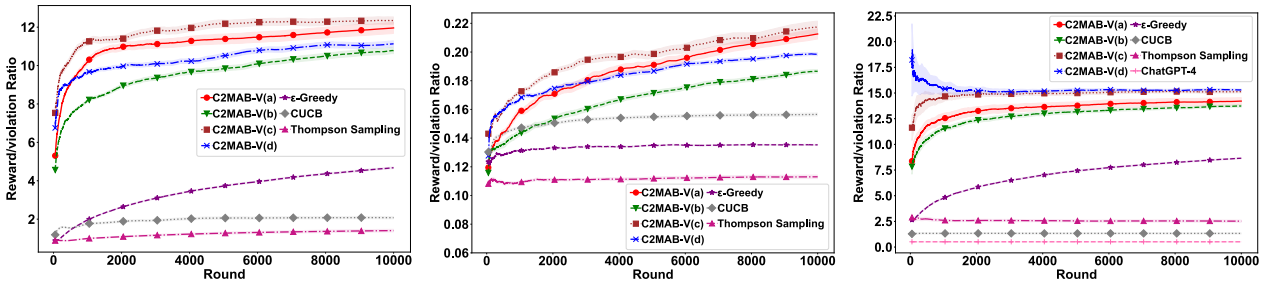
A. Application Settings and Dataset Description

1) Mobile Crowdsensing: In mobile crowdsensing, we experiment with simulated data. Consider $L = 25$ sensing tasks, where an agent selects $K = 4$ tasks for users to perform. Each task i has a fixed reward $p_i = 0.5$, an unknown participation probability $b_i \in [0,1]$ (based on user engagement), and an unknown resource cost $c_i \in [0,1]$ for task execution. The expected reward is $w_i = p_i b_i$. The agent knows p_i but not b_i or c_i. We simulate b_i and c_i from a uniform distribution $U[0,1]$. At round t, after selecting tasks $\boldsymbol{S}_t$, the environment generates a Bernoulli variable $\boldsymbol{b}_{t,i} \in \{0,1\}$ with mean b_i (indicating user participation) and a cost from a truncated normal distribution $\mathcal{N}(c_i, 1)$ on $[c_i - d_i, c_i + d_i]$, where $d_i = \min\{c_i, 1-c_i\}$. The agent earns reward $\hat{f}_1(\boldsymbol{S}_t, \boldsymbol{w}_t) = \sum_{i\in\boldsymbol{S}_t} \boldsymbol{w}_{t,i}$, incurs cost $\hat{g}(\boldsymbol{S}_t, \boldsymbol{c}_t) = \sum_{i\in\boldsymbol{S}_t} \boldsymbol{c}_{t,i}$, and observes semi-bandit feedback: $\boldsymbol{w}_{t,i} = p_i \boldsymbol{b}_{t,i}$ and $\boldsymbol{c}_{t,i}$ for $i \in \boldsymbol{S}_t$. The process runs for $T = 200,000$ rounds with a cost constraint $b = 1.4$.

2) Network Routing: We experiment with the RocketFuel dataset [56], where we choose Network 1755 of it that has 300 nodes and 549 edges. We preprocess data by calculating betweenness centrality for each node, representing its functional importance [57]. Two pairs of nodes are randomly selected as start and end points for routing. For each pair, we identify all simple paths, select nodes on these paths as candidates, and normalize their betweenness centrality. We retain 30 nodes with values between 0.05 and 0.95 as the ground set $\mathcal{E}$, repeating this 5 times per pair for 10 random experiments. Node capacity correlates with importance, so we set $w_i = \beta \cdot b_i$ and $c_i = \gamma \cdot b_i$, where b_i is the normalized betweenness centrality, and $\beta = \gamma = 1$, with a cost constraint $b = 1.6$.

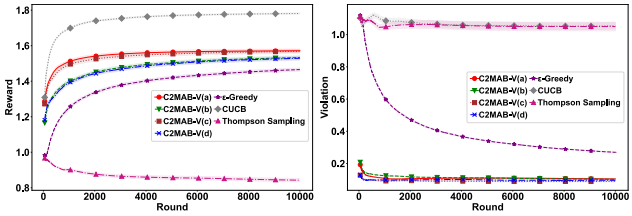
3) Multi-LLM Selection: We conduct experiments for multi-LLM selection using the SciQ dataset [58], which spans topics such as physics, chemistry, and biology, across nine LLMs detailed in Table IV. Costs are derived from official pricing, with a maximum selection of $K = 4$ LLMs and a budget threshold $b = 0.495$. The agent aims to select up to $K = 4$ LLMs to maximize the probability that at least one LLM correctly answers a question. At round t, the agent selects a subset $\boldsymbol{S}_t$ and receives a reward $\hat{f}_3(\boldsymbol{S}_t, \boldsymbol{w}_t) = 1 - \prod_{i\in\boldsymbol{S}_t}(1-\boldsymbol{w}_{t,i})$. Besides the baselines mentioned, we compare performance against consistently using the high-cost ChatGPT-4 [59].

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on July 16,2026 at 05:50:42 UTC from IEEE Xplore. Restrictions apply.

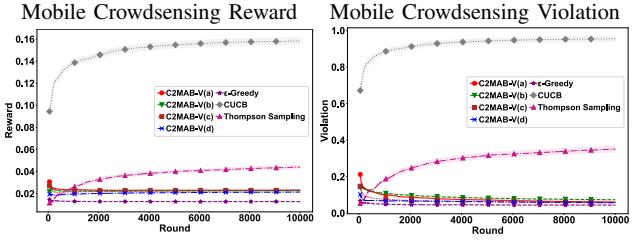


(a) Mobile Crowdsensing (b) Network Routing (c) Multi-LLM Selection

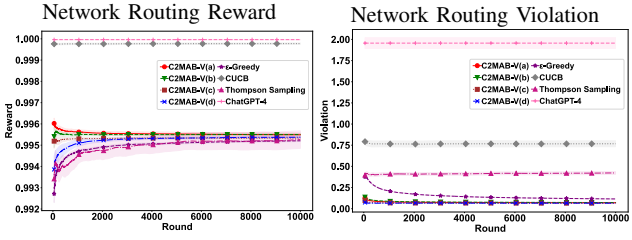
Fig. 2: Reward/violation ratio of the three applications, with detailed results for individual reward and violation in Fig. 3.



Mobile Crowdsensing Reward Mobile Crowdsensing Violation



Network Routing Reward Network Routing Violation



Multi-LLM Selection Reward Multi-LLM Selection Violation

Fig. 3: Per-round reward and violation of three applications.

TABLE IV: List of utilized LLM APIs with cost per usage.

Model Name (with Parameters)	**Cost** (USD/1k tokens)	**Size** (GB)
ChatGLM2-6B-32K [59]	0.005	12.5
ChatGPT-3.5 [59]	0.02	/
Claude 2 [60]	0.08	/
ERNIE 3.5-8K [61]	0.015	/
Llama 2-7B [62]	0.005	12.6
Llama 2-13B [62]	0.008	24.3
Llama 2-70B [62]	0.05	128.3
Mixtral-8x7B-Instruct [63]	0.05	93.37
ChatGPT-4 [59]	0.12	/

B. Evaluation Results

The reward/violation ratio performance is shown in Fig. 2. In the multi-LLM selection scenario (Fig. 2(c)), C2MAB-V consistently outperforms all other algorithms across four parameter settings, achieving the highest reward/violation ratio. This highlights C2MAB-V's superior ability to balance rewards and constraint violations. Our algorithm converges in under 1,000 rounds, significantly faster than the ϵ-Greedy baseline, which requires nearly 10,000 rounds, even without leveraging offline information and relying solely on online learning. Similar trends are observed in the other two applications (Figs. 2(a) and 2(b)), further demonstrating C2MAB-V's robustness.

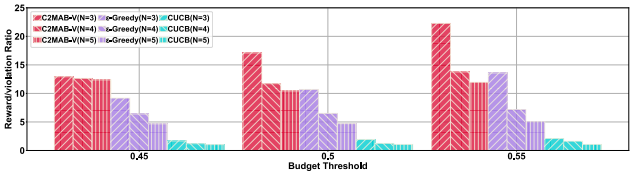


Fig. 4: Impact of different budget thresholds b and maximum number K over 3000 rounds.

We then examine the impact of the budget threshold b and the maximum number of selections K in the multi-LLM selection problem. As shown in Fig. 4, the optimal combination of LLMs varies with changes in b and K. For a fixed budget threshold b, increasing K expands the selection space, which increases complexity and impacts the efficiency of all algorithms. However, as K increases, C2MAB-V consistently outperforms the CUCB and ϵ-Greedy baselines. Moreover, across various values of b, C2MAB-V achieves at least 356.0% improvement over CUCB and 55.7% over ϵ-Greedy, demonstrating its superior robustness and performance.

VIII. Conclusion and Future Directions

This paper studies the constraint-aware CMAB problem, integrating relaxation and rounding techniques to propose C2MAB-V, a novel framework designed to address this challenge. Proposing the concept of *nice RR conditions*, we demonstrate that C2MAB-V achieves sublinear regret, rapidly diminishing constraint violations, and polynomial time complexity. We present three practical applications meeting these conditions, with C2MAB-V exhibiting superior performance in experiments, including empirical evaluations across nine LLMs for the multi-LLM selection problem. Future work will focus on refining the regret analysis to eliminate the p^* coefficient and exploring adversarial or contextualized reward/cost structures.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on July 16,2026 at 05:50:42 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[2] J. Cheng, N. Ding, J. C. Lui, and J. Huang, "Continuous query-based data trading," in *Proc. ACM SIGMETRICS*, 2024, pp. 73–74.

[3] X. Liu, J. Zuo, J. Wang, Z. Wang, Y. Xu, and J. C. Lui, "Learning context-aware probabilistic maximum coverage bandits: A variance-adaptive approach," in *Proc. IEEE INFOCOM*. IEEE, 2024.

[4] W. Chen, Y. Wang, Y. Yuan, and Q. Wang, "Combinatorial multi-armed bandit and its extension to probabilistically triggered arms," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1746–1778, 2016.

[5] K. Cai, X. Liu, Y.-Z. J. Chen, and J. C. Lui, "An online learning approach to network application optimization with guarantee," in *Proc. IEEE INFOCOM*. IEEE, 2018, pp. 2006–2014.

[6] X. Liu, B. Li, P. Shi, and L. Ying, "Pond: Pessimistic-optimistic online dispatching," *arXiv:2010.09995*, 2020.

[7] T. Liu, R. Zhou, D. Kalathil, P. Kumar, and C. Tian, "Learning policies with zero or bounded constraint violation for constrained mdps," *Proc. NeurIPS*, vol. 34, pp. 17 183–17 193, 2021.

[8] X. Liu, B. Li, P. Shi, and L. Ying, "An efficient pessimistic-optimistic algorithm for stochastic linear bandits with general constraints," *Proc. NeurIPS*, vol. 34, pp. 24 075–24 086, 2021.

[9] A. Badanidiyuru, R. Kleinberg, and A. Slivkins, "Bandits with knapsacks," in *Proc. IEEE FOCS*. IEEE, 2013, pp. 207–216.

[10] S. Agrawal and N. R. Devanur, "Bandits with global convex constraints and objective," *Oper. Res.*, vol. 67, no. 5, pp. 1486–1502, 2019.

[11] J. Zuo and C. Joe-Wong, "Combinatorial multi-armed bandits for resource allocation," in *Proc. CISS*. IEEE, 2021, pp. 1–4.

[12] S. Li, B. Wang, S. Zhang, and W. Chen, "Contextual combinatorial cascading bandits," in *Proc. ICML*. PMLR, 2016, pp. 1245–1253.

[13] Q. Wang and W. Chen, "Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications," *Proc. NeurIPS*, vol. 30, 2017.

[14] K. A. Sankararaman and A. Slivkins, "Combinatorial semi-bandits with knapsacks," in *Proc. AISTATS*. PMLR, 2018, pp. 1760–1770.

[15] K. Chen, K. Cai, L. Huang, and J. C. Lui, "Beyond the click-through rate: web link selection with multi-level feedback," in *Proc. IJCAI*, 2018, pp. 3308–3314.

[16] S. Takemori, M. Sato, T. Sonoda, J. Singh, and T. Ohkuma, "Submodular bandit problem under multiple constraints," in *Proc. UAI*. PMLR, 2020, pp. 191–200.

[17] S. Bubeck, N. Cesa-Bianchi *et al.*, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

[18] A. Slivkins *et al.*, "Introduction to multi-armed bandits," *Found. Trends Mach. Learn.*, vol. 12, no. 1-2, pp. 1–286, 2019.

[19] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1466–1478, 2012.

[20] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari, "Tight regret bounds for stochastic combinatorial semi-bandits," in *Proc. AISTATS*, 2015.

[21] R. Combes, M. S. Talebi Mazraeh Shahi, A. Proutiere *et al.*, "Combinatorial bandits revisited," in *Proc. NeurIPS*, vol. 28, 2015.

[22] B. Kveton *et al.*, "Cascading bandits: Learning to rank in the cascade model," in *Proc. ICML*. PMLR, 2015, pp. 767–776.

[23] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvári, "Combinatorial cascading bandits," in *Proc. NeurIPS*, vol. 28, 2015, pp. 1450–1458.

[24] Z. Wen, B. Kveton, M. Valko, and S. Vaswani, "Online influence maximization under independent cascade model with semi-bandit feedback," in *Proc. NeurIPS*, vol. 30, 2017.

[25] B. Kveton *et al.*, "Matroid bandits: fast combinatorial optimization with learning," in *Proc. UAI*, 2014, pp. 420–429.

[26] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *Proc. ICML*. PMLR, 2013, pp. 151–159.

[27] W. Ding, T. Qin, X.-D. Zhang, and T.-Y. Liu, "Multi-armed bandit with budget constraint and variable costs," in *Proc. AAAI*, 2013.

[28] H. Wu, R. Srikant, X. Liu, and C. Jiang, "Algorithms with logarithmic or sublinear regret for constrained contextual bandits," in *Proc. NeurIPS*, vol. 28, 2015, pp. 433–441.

[29] Y. Xia, T. Qin, W. Ma, N. Yu, and T.-Y. Liu, "Budgeted multi-armed bandits with multiple plays," in *Proc. IJCAI*, 2016, pp. 2210–2216.

[30] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," in *Proc. IEEE INFOCOM*. IEEE, 2019, pp. 1702–1710.

[31] D. P. Williamson and D. B. Shmoys, *The design of approximation algorithms*. Cambridge Univ. Press, 2011.

[32] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," *J. ACM*, vol. 53, no. 3, pp. 324–360, 2006.

[33] C. Chekuri, J. Vondrák, and R. Zenklusen, "Dependent randomized rounding for matroid polytopes and applications," *arXiv:0909.4348*, 2009.

[34] P. M. Vaidya, "Speeding-up linear programming using fast matrix multiplication," in *Proc. IEEE FOCS*, 1989, pp. 332–337.

[35] D. Harris, T. Pensyl, A. Srinivasan, and K. Trinh, "Dependent randomized rounding for clustering and partition systems with knapsack constraints," in *Proc. AISTATS*. PMLR, 2020, pp. 2273–2283.

[36] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a submodular set function subject to a matroid constraint," in *Proc. IPCO*. Springer, 2007, pp. 182–196.

[37] Y. Bian, J. M. Buhmann, and A. Krause, "Continuous submodular function maximization," *arXiv:2006.13474*, 2020.

[38] X. Dai, X. Sun, J. Zuo, X. Liu, and J. C. Lui, "A unified online-offline framework for co-branding campaign recommendations," in *Proc. ACM SIGKDD*, 2025, pp. 427–438.

[39] X. Dai, X. Liu, J. Zuo, H. Xie, C. Joe-Wong, and J. C. S. Lui, "Variance-aware bandit framework for dynamic probabilistic maximum coverage problem with triggered or self-reliant arms," *IEEE/ACM Trans. Netw.*, pp. 1–12, 2025.

[40] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: online convex optimization with long term constraints," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 2503–2528, 2012.

[41] K. Cai, X. Liu, Y.-Z. J. Chen, and J. C. Lui, "Learning with guarantee via constrained multi-armed bandit: Theory and network applications," *IEEE Trans. Mob. Comput.*, 2022.

[42] T. M. Chan, "Approximation schemes for 0-1 knapsack," in *Proc. SOSA*, 2018.

[43] X. Dai, Y. Xie, M. Liu, X. Wang, Z. Li, H. Wang, and J. C. S. Lui, "Multi-agent conversational online learning for adaptive llm response identification," *arXiv:2501.01849*, 2025.

[44] Y. Yang, Y. Tang, and K. Y. Tam, "Investlm: A large language model for investment using financial domain instruction tuning," *arXiv:2309.13064*, 2023.

[45] L. Chen, M. Zaharia, and J. Zou, "Frugalgpt: How to use large language models while reducing cost and improving performance," *arXiv:2305.05176*, 2023.

[46] X. Dai, J. Li, X. Liu, A. Yu, and J. Lui, "Cost-effective online multi-llm selection with versatile reward models," *arXiv:2405.16587*, 2024.

[47] Poe, "Poe," https://poe.com/ChatGPT, 2025.

[48] Z. Liu, Y. Zhang, P. Li, Y. Liu, and D. Yang, "Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization," *arXiv:2310.02170*, 2023.

[49] Awan-LLM, "Awan," https://www.awanllm.com/, 2025.

[50] JD-Cloud-Yanxi-AI, "JDCloud," https://yanxi.jd.com/, 2025.

[51] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma, "An explanation of in-context learning as implicit bayesian inference," *arXiv:2111.02080*, 2021.

[52] S. Dalal and V. Misra, "The matrix: A bayesian learning model for llms," *arXiv:2402.03175*, 2024.

[53] Gurobi-Optimization, "GUROBI," https://www.gurobi.com/, 2025.

[54] X. Liu, X. Dai, X. Wang, M. Hajiesmaili, and J. C. Lui, "Combinatorial logistic bandits," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 53, no. 1, pp. 112–114, 2025.

[55] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, pp. 235–256, 2002.

[56] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 133–145, 2002.

[57] S. Dolev, Y. Elovici, and R. Puzis, "Routing betweenness centrality," *J. ACM*, vol. 57, no. 4, pp. 1–27, 2010.

[58] J. Welbl, N. F. Liu, and M. Gardner, "Crowdsourcing multiple choice science questions," *arXiv:1707.06209*, 2017.

[59] OpenAI, "OpenAI LLM API," https://platform.openai.com/, 2025.

[60] Claude, "Claude AI LLM API," https://www.anthropic.com/, 2025.

[61] Baidu, "Wenxin," https://wenxin.baidu.com/, 2025.

[62] Ollama, "Ollama," https://github.com/jmorganca/ollama/, 2025.

[63] Mistral-AI, "Mistral," https://mistral.ai/, 2025.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on July 16,2026 at 05:50:42 UTC from IEEE Xplore. Restrictions apply.