QUANTUM ALGORITHMS FOR BANDITS WITH KNAPSACKS WITH IMPROVED REGRET AND TIME COMPLEXITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Bandits with knapsacks (BwK) is a fundamental model that combines stochastic integer programming with online learning. Classical algorithms for BwK with a time horizon T achieve a problem-independent regret bound of $\mathcal{O}(\sqrt{T})$ and a problem-dependent bound of $\mathcal{O}(\log T)$. In this paper, we initiate the study of the BwK model in the setting of quantum computing, where both reward and resource consumption can be accessed via quantum oracles. We establish both problem-independent and problem-dependent regret bounds for quantum BwK algorithms. For the problem-independent case, we demonstrate that a quantum approach can improve the classical regret bound by a factor of $(1 + \sqrt{B/\mathrm{OPT}_{\mathrm{LP}}})$, where B is the budget constraint in BwK and $\mathrm{OPT}_{\mathrm{LP}}$ denotes the optimal value of a linear programming relaxation of the BwK problem. For the problem-dependent setting, we develop a quantum algorithm using an inexact quantum linear programming solver. This algorithm achieves a quadratic improvement in terms of the problem-dependent parameters, as well as a polynomial speedup of time complexity on the problem's dimensions compared to classical counterparts. Compared to previous works on quantum algorithms for multi-armed bandits, our study is the first to consider bandit models with resource constraints, extending quantum speedups to resource-constrained online learning.

1 INTRODUCTION

The multi-armed bandit (MAB) problem is a fundamental model in machine learning and has been widely studied in online decision-making and reinforcement learning (Slivkins, 2019; Gittins et al., 2011). First introduced by Robbins (1952), the MAB problem involves a decision-maker faced with a fixed set of arms, each associated with an unknown stochastic reward. During the online decision-making process, repeated selection of an arm yields increasing information about its reward distribution. The objective in the MAB problem is to minimize the cumulative regret over a fixed time horizon T, a goal that necessitates navigating the exploration-exploitation trade-off (Weber, 1992). A significant extension to this framework is the Bandits with Knapsacks (BwK) problem, which introduces resource constraints into the arm-selection process, drawing on ideas from the classical knapsack problem (Badanidiyuru et al., 2018). This model lies at the intersection of online linear programming (LP) and Markov decision processes (MDPs) (Yu et al., 2009; Borkar & Meyn, 2002; Wang, 2020; Bertsimas & Niño-Mora, 2000; Ruszczyński, 2010), and has also been connected to a wide range of real-world problems in online learning, including dynamic pricing and revenue management (Besbes & Zeevi, 2012; Ferreira et al., 2018), online advertising (Mehta et al., 2007), and network routing (Agrawal et al., 2014).

Quantum computing offers new possibilities for solving optimization problems by using quantum effects to explore complex solution spaces more efficiently than classical approaches. Techniques such as superposition and entanglement allow quantum algorithms to solve certain tasks faster. Within the domain of constrained optimization, quantum algorithms have demonstrated a polynomial speedup compared to their classical counterpart (Grigoriadis & Khachiyan, 1995) in solving zero-sum games and linear programming (LP) problems through fast quantum multi-Gibbs sampling (Gao et al., 2024; Bouland et al., 2023). There also exist fast quantum subroutines for the simplex method (Nannicini, 2024). More general, semi-definite programming (SDP) also constitutes

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

a significant class within constrained optimization, and quantum SDP solvers (Brandão & Svore, 2017; Brandão et al., 2019; Apeldoorn et al., 2017; Apeldoorn & Gilyén, 2019b; Augustino et al., 2023b) offer a polynomial speedup over the current state-of-the-art classical SDP solvers (Lee et al., 2015). Furthermore, polynomial speedup is also proved for general constrained convex optimization problems (Chakrabarti et al., 2020; Apeldoorn et al., 2020). Beyond constrained optimization, the potential of quantum algorithms also extends to unconstrained optimization, especially nonconvex optimization (Gong et al., 2025; Liu et al., 2023; Zhang & Li, 2023; Chen et al., 2025; Leng et al., 2023; 2025). A general survey of quantum algorithms for optimization can be found in Abbas et al. (2024).

On the other hand, the intersection of quantum computing and reinforcement learning (RL) has yielded notable advancements (Dunjko & Briegel, 2018; Meyer et al., 2022; Jerbi et al., 2022). For instance, Wang et al. (2021) introduced a quantum exploration algorithm for efficiently identifying the optimal arm in a MAB setting, achieving a quadratic quantum speedup compared to the best possible classical result.Wan et al. (2023) investigated quantum multi-armed bandits and stochastic linear bandits, establishing a regret bound of $\mathrm{poly}(\log T)$ regret for both models, thus achieving an exponential speedup compared to the $\Omega(\sqrt{T})$ regret bound of classical MAB algorithms (Auer et al., 2002). This $\mathrm{poly}(\log T)$ regret was subsequently extended to the quantum version of kernelized bandits (Dai et al., 2023). Other quantum bandit settings have also been studied (Wu et al., 2023; Siam et al., 2026). More recently, Zhong et al. (2023) studied the online exploration problem in quantum RL, presenting a novel quantum algorithm for both tabular MDPs and linear mixture MDPs with poly-logarithmic regrets.

While these theoretical developments mark significant progress, applying quantum algorithms to real-world problems—particularly those arising at the intersection of constrained optimization and online learning—remains a largely open and challenging area of research. Among practical settings in online learning, decision-making agents must simultaneously learn from data and operate under strict resource constraints, necessitating algorithmic frameworks that can handle both aspects effectively.

Main results In this paper, we investigate the BwK problem in the quantum computing setting. Our model consists of m arms and d resources, operates over a finite time horizon T, and is subject to a total resource budget B. We develop two primal-dual quantum algorithms that achieve improved regret bounds in the problem-independent and problem-dependent settings. The problem-independent algorithm combines quantum reward estimation with per-round resource updates to obtain a general guarantee in terms of m, d, T, and B. The problem-dependent algorithm uses quantum exploration to identify optimal arms and non-binding constraints, followed by adaptive resource allocation based on approximate quantum LP solutions. This design yields a finer regret bound in terms of the instance parameters defined in Section 2.1, together with improved computational complexity for LP computations in both phases. Our algorithms build on the classical frameworks of Badanidiyuru et al. (2018) and Li et al. (2021), respectively. Table 1 summarizes the expected regret bounds and compares them with their classical counterparts.

Table 1: Comparison of regret bounds for classical and quantum algorithms for BwK problems.

Model	Method	Setting	Regret
Problem Independent	Badanidiyuru et al. (2018)	Classical	$\mathcal{O}\left(\sqrt{\log(dT)}\right)\left(\mathrm{OPT}_{\mathrm{LP}}\sqrt{\frac{m}{B}}\left(1+\sqrt{\frac{B}{\mathrm{OPT}_{\mathrm{LP}}}}\right)\right)$ $+\mathcal{O}(m\log(dT)\log(T))$
	Theorem 1	Quantum	$\mathcal{O}\left(\sqrt{\log(dT)}\right)\left(\mathrm{OPT}_{\mathrm{LP}}\sqrt{\frac{m}{B}}\right)+\mathcal{O}(m\log(dT)\log(T))$
Problem Dependent	Li et al. (2021)	Classical	$\mathcal{O}\left(\left(2+\frac{1}{b}\right)^2\frac{md}{b\delta^2}\log T+\frac{d^4}{b^2\min\{\chi^2,\delta^2\}\min\{1,\sigma^2\}}\right)$
	Theorem 3	Quantum	$\tilde{\mathcal{O}}\left(\left(2+\frac{1}{b}\right)\frac{m\sqrt{d}}{b\delta}\log T+\frac{d^4}{b^2\min\{\chi^2,\delta^2\}\min\{1,\sigma^2\}}\right)$[1]

In the problem-independent regime, our quantum algorithm improves the leading regret term by a factor of $(1+\sqrt{B/\mathrm{OPT}_{\mathrm{LP}}})$ over the classical counterpart. In the problem-dependent regime, our

[1]In this paper, we use the notation $\tilde{\mathcal{O}}$ to suppress poly-log factors, i.e., $\tilde{\mathcal{O}}(f)=\mathcal{O}(f\,\mathrm{poly}(\log f))$ for a function f.

quantum algorithm yields a quadratic improvement in the coefficient of the leading $\log T$ term with respect to $1/\delta$ (with δ defined in Eq. (6)) and $(2 + \frac{1}{b})$, where $b = B/T$, and reduces its dimension dependence from d to $\sqrt{d}$.

Beyond regret, our problem-dependent algorithm also addresses the computational bottleneck of repeatedly solving LP problems in BwK. We combine quantum exploration with adaptive resource allocation and establish that coarse approximate LP solutions suffice to achieve the improved regret guarantees. This design enables fast quantum LP solvers (Gao et al., 2024; Bouland et al., 2023) to accelerate the LP computations throughout the algorithm. As summarized in Table 2, our algorithm achieves a polynomial speedup in its dependence on the numbers of arms m and resources d compared with the classical problem-dependent algorithm of Li et al. (2021).

Table 2: Time complexities of classical and quantum problem-dependent algorithms for BwK.

Algorithm	**Time complexity**
Li et al. (2021)	$\tilde{\mathcal{O}}\left(\max\{m,d\}^{2.372}(T+m+d)\right)$
Algorithm 2	$\tilde{\mathcal{O}}\left(md + \frac{(m+d)^{1.5}}{\epsilon_{\mathrm{LP}}^{2.5}} + \left(\frac{\sqrt{m+d}}{\epsilon_{\mathrm{LP}}^{2.5}} + d\right)T\right)$

Techniques In our quantum algorithm, we leverage the quantum Monte Carlo method (QMC) to obtain tighter upper and lower confidence bounds (UCB and LCB) on random variables. Intuitively, after N samples the confidence-interval length in classical Monte Carlo methods scales as $\mathcal{O}(1/\sqrt{N})$, whereas in the quantum Monte Carlo method it improves to $\tilde{\mathcal{O}}(1/N)$ via quantum amplitude estimation. However, QMC yields useful information only upon measurement—and each measurement collapses the quantum state—so we cannot reuse information from previous samples within the same quantum run, unlike in classical Monte Carlo methods. Consequently, we design a geometric sampling schedule for both our problem-independent and problem-dependent algorithms, in which successive QMC executions use geometrically increasing query budgets to tighten the UCB and LCB estimates that guide primal-dual decisions.

Beyond using QMC to improve statistical estimation, we further develop a quantum-accelerated problem-dependent algorithm by addressing its LP-solving bottleneck. The key challenge is to achieve accurate identification and efficient resource utilization using coarse approximate LP solutions. We address this by designing adaptive LP formulations and establishing phase-specific accuracy requirements that control how approximation errors affect identification decisions and the evolution of the remaining resource budget.

Our algorithm has two phases. In the identification phase, we compare confidence-bound LP values to identify optimal arms $\mathcal{I}^*$ and non-binding constraints $\mathcal{J}'$. By controlling the combined quantum estimation and LP optimization errors below the separation gap, we show that solving each LP only to additive accuracy $\epsilon_{\mathrm{LP}}T$ suffices for correct identification in Proposition 3. In the adaptive allocation phase, we add lower constraints on estimated binding-resource consumption when the resource estimates are accurate and the average remaining budget $\boldsymbol{B}^{(t)}/(T-t)$ at time t is close to its initial level $\boldsymbol{B}/T$. These additional constraints control the perturbation caused by approximate LP solutions and ensure that the probability of selecting each optimal arm is lower bounded, even when the LP is solved only to additive accuracy $\epsilon_{\mathrm{LP}}(T-t)/\log^2 T$. This guarantees sufficient sampling of every optimal arm. A martingale argument then shows that the expected unused binding resources at termination are bounded by a problem-dependent constant $\mathcal{O}(1)$ in Proposition 4. Together, these techniques yield a polynomial improvement in time complexity.

2 Preliminaries

We consider the BwK problem with m arms, d resources, and a finite time horizon T. In each round $t \in [T]$, the algorithm selects an arm $i \in [m]$, receives a reward $r_t \in [0,1]$, and consumes a resource vector $\boldsymbol{c}_t \in [0,1]^d$. Each resource $j \in [d]$ is subject to a budget constraint $B_j \in \mathbb{R}^+$, and the algorithm terminates either when T rounds have been completed or when any resource budget is exceeded. For each arm $i \in [m]$, the reward and cost vectors are i.i.d. drawn from an unknown distribution $\boldsymbol{\pi}_i$ supported on $[0,1]^{d+1}$. We denote the expected reward vector as $\boldsymbol{r} \in [0,1]^m$ and the

expected cost matrix as $\boldsymbol{C} \in [0,1]^{d\times m}$, with $\boldsymbol{C}_{\cdot,i}$ and $\boldsymbol{C}_{j,\cdot}$ denoting the i-th column and the j-th row, respectively. Let OPT denote the maximum achievable expected reward under the resource constraints. The *expected regret* of an algorithm is OPT minus its expected total reward.

Without loss of generality, we assume that all resource budgets are uniform and no larger than the time horizon, i.e., $B_j = B \le T$ for all $j \in [d]$. We define $b = B/T \in [0,1]$ and $\boldsymbol{B} = (B, B, \ldots, B)^\top$. For analytical convenience, we incorporate the time horizon T as the first resource constraint by setting $C_{1,i} = b$ for all $i \in [m]$.

2.1 Primal-dual perspective for BwK

LP relaxation of optimal value Let $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_m) \in \mathbb{R}^m$, where ξ_i is the expected number of pulls assigned to arm i. The primal and dual LPs share the optimal value $\mathrm{OPT}_{\mathrm{LP}}$:

$$\max_{\boldsymbol{\xi}\ge\mathbf{0},\, \boldsymbol{C}\boldsymbol{\xi}\le\boldsymbol{B}} \boldsymbol{r}^\top\boldsymbol{\xi} \quad \text{(primal LP)}, \qquad (1a) \qquad\qquad \min_{\boldsymbol{\eta}\ge\mathbf{0},\, \boldsymbol{C}^\top\boldsymbol{\eta}\ge\boldsymbol{r}} \boldsymbol{B}^\top\boldsymbol{\eta} \quad \text{(dual LP)}. \qquad (1b)$$

We denote the primal and dual optimal solutions as $\boldsymbol{\xi}^*$ and $\boldsymbol{\eta}^*$, respectively. Since this relaxation ignores online adaptivity and allocates arms fractionally, by Badanidiyuru et al. (2018, Lemma 3.1), $\mathrm{OPT}_{\mathrm{LP}}$ upper bounds the true optimal reward OPT of BwK. The dual variables can be viewed as resource prices: $\boldsymbol{C}^\top\boldsymbol{\eta} \ge \boldsymbol{r}$ requires every arm to be paid for by the resources it consumes, and $\boldsymbol{B}^\top\boldsymbol{\eta}$ is the total value of the available budget.

Arm-bandit symmetry We partition the arms into optimal arms $\mathcal{I}^*$ and sub-optimal arms $\mathcal{I}'$, and the constraints into binding constraints $\mathcal{J}^*$ and non-binding constraints $\mathcal{J}'$ according to

$$\begin{aligned} \mathcal{I}^* &:= \{\xi_i^* > 0, i \in [m]\}, & \mathcal{I}' &:= \{\xi_i^* = 0, i \in [m]\}, \\ \mathcal{J}^* &:= \{B - \sum_{i=1}^m C_{ji}\xi_i^* = 0,\ j \in [d]\}, & \mathcal{J}' &:= \{B - \sum_{i=1}^m C_{ji}\xi_i^* > 0,\ j \in [d]\}. \end{aligned} \qquad (2)$$

Here $\mathcal{I}^*$ contains the arms used by the optimal relaxed policy, and $\mathcal{J}^*$ contains the resources fully spent under this policy. These two sets identify the active part of the instance: where reward is collected and which resources limit the achievable reward. We use the following standard uniqueness and non-degeneracy assumption, which can be satisfied through an arbitrarily small perturbation (Megiddo & Chandrasekaran, 1989).

Assumption 1. $\mathrm{OPT}_{\mathrm{LP}}$ *in (1a) has a unique optimal solution, and the optimal solution is non-degenerate:* $|\mathcal{I}^*| = |\mathcal{J}^*|$.

Define the LP problems OPT_i and OPT_j by removing arm i or modifying constraint j from the original LP:

$$\mathrm{OPT}_i := \max_{\boldsymbol{\xi}}\{\boldsymbol{r}^\top\boldsymbol{\xi} : \boldsymbol{C}\boldsymbol{\xi} \le \boldsymbol{B},\ \xi_i = 0,\ \boldsymbol{\xi} \ge \mathbf{0}\}, \qquad (3)$$

$$\mathrm{OPT}_j := \min_{\boldsymbol{\eta}}\{\boldsymbol{B}^\top\boldsymbol{\eta} - B : \boldsymbol{C}^\top\boldsymbol{\eta} \ge \boldsymbol{r} + \boldsymbol{C}_{j,\cdot},\ \boldsymbol{\eta} \ge 0\}. \qquad (4)$$

Intuitively, OPT_i tests whether arm i is indispensable, and OPT_j probes the sensitivity of the dual value to resource j. Proposition 1 characterizes these effects in terms of optimality gaps.

Proposition 1 (Li et al. (2021, Proposition 2))**.** *With Assumption 1, the following inequality holds:*

$$\begin{aligned} &\mathrm{OPT}_i < \mathrm{OPT}_{\mathrm{LP}} \Leftrightarrow i \in \mathcal{I}^*, \quad \mathrm{OPT}_i = \mathrm{OPT}_{\mathrm{LP}} \Leftrightarrow i \in \mathcal{I}', \\ &\mathrm{OPT}_j = \mathrm{OPT}_{\mathrm{LP}} \Leftrightarrow j \in \mathcal{J}^*, \quad \mathrm{OPT}_j < \mathrm{OPT}_{\mathrm{LP}} \Leftrightarrow j \in \mathcal{J}'. \end{aligned} \qquad (5)$$

Notations for problem-dependent BwK To ensure consistency with prior work on problem-dependent BwK, we adopt the notations introduced in Li et al. (2021). Given $\mathcal{I}^*$ and $\mathcal{J}^*$, define

$$\begin{aligned} &\delta := \frac{1}{T}(\mathrm{OPT}_{\mathrm{LP}} - \max\{\max_{i\in\mathcal{I}^*}\mathrm{OPT}_i, \max_{j\in\mathcal{J}'}\mathrm{OPT}_j\}), \\ &\sigma := \sigma_{\min}\left(\boldsymbol{C}_{\mathcal{J}^*,\mathcal{I}^*}\right),\ \chi := \frac{1}{T}\min\{\xi_i^* \ne 0, i \in [m]\},\ s := |\mathcal{I}^*| = |\mathcal{J}^*|. \end{aligned} \qquad (6)$$

Here, δ denotes the separation gap between the optimal LP value and the best sub-optimal LP value, σ captures the minimum singular value of the active resource-arm submatrix, χ denotes the smallest nonzero component of the optimal primal solution, and s is the size of the optimal arms set.

2.2 QUANTUM ORACLES AND MEAN ESTIMATION

We first introduce the quantum notation used throughout the paper. An n-qubit pure quantum state is a unit vector $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle \in \mathbb{C}^{2^n}$, where $\{|x\rangle\}_{x\in\{0,1\}^n}$ denotes the computational basis and $\sum_x |\alpha_x|^2 = 1$. Measuring $|\psi\rangle$ in the computational basis returns x with probability $|\alpha_x|^2$. For two quantum states $|u\rangle$ and $|v\rangle$, we write $|u\rangle|v\rangle := |u\rangle \otimes |v\rangle$ for their tensor product. Quantum operations are represented by unitary operators; for a unitary $\mathcal{O}$, we denote its inverse by $\mathcal{O}^\dagger$.

We use quantum sampling oracles to access the reward and resource consumption distributions and construct the corresponding UCB and LCB estimates. For arm i, let Ω_i^r and $\Omega_i^{\boldsymbol{C}}$ denote the finite sample spaces of its reward and resource consumption, respectively.

Definition 1 (Quantum oracles for BwK)**.**

$$\mathcal{O}_i^a : |0\rangle \mapsto \sum_{\omega\in\Omega_i^a} \sqrt{P_i^a(\omega)}\,|\omega\rangle\,|y_i^a(\omega)\rangle\,, \qquad a \in \{r, \boldsymbol{C}\}, \tag{7}$$

where P_i^a is the corresponding probability distribution, $y_i^r : \Omega_i^r \to [0,1]$ is the reward, and $y_i^{\boldsymbol{C}} : \Omega_i^{\boldsymbol{C}} \to [0,1]^d$ is the resource consumption vector.

This coherent oracle model extends classical sampling. Measuring $\mathcal{O}_i^a|0\rangle$ in the computational basis immediately after the oracle call returns a classical sample from P_i^a. A classical sample can be stored and reused, allowing the empirical mean to be updated after each new sample. In quantum mean estimation, the algorithm performs coherent operations over multiple oracle calls without observing individual samples, and obtains a classical estimate only after the final measurement. This measurement collapses the coherent state, and preparing another coherent state requires a new oracle invocation. Following prior quantum-bandit work (Wan et al., 2023), each use of $\mathcal{O}_i^a$ or $(\mathcal{O}_i^a)^\dagger$ is counted as one query, or equivalently one pull, of arm i. See Appendix A.1 for further details on quantum sampling and mean estimation.

We now recall the mean-estimation guarantees under classical sampling and coherent quantum-oracle access. For classical sampling, the UCB and LCB estimators satisfy the following guarantee by the Chernoff–Hoeffding inequality.

Lemma 1 (Classical univariate and multivariate mean estimator)**.** *Let $\boldsymbol{y} : \Omega \to [0,1]^d$ be a d-dimensional random variable with bounded value, and let $\hat{\boldsymbol{y}}$ be the average of N independent samples from this distribution. Then the Chernoff-Hoeffding's inequality guarantees that $P(|\hat{\boldsymbol{y}} - \mathbb{E}[\boldsymbol{y}]| \le \epsilon) \ge 1-\delta$ with at most $\frac{\log(2d/\delta)}{\epsilon^2}$ samples.*

In the quantum setting, QMC based on amplitude estimation (Brassard et al., 2002) gives the following univariate and multivariate estimators.

Lemma 2 (Quantum univariate mean estimator (Montanaro, 2015; Kothari & O'Donnell, 2023))**.** *Assume that $y : \Omega \to [0,1]$ is a random variable, Ω is equipped with a probability measure P, and the quantum unitary oracle $\mathcal{O}$ encodes P and y. There is a constant $C_1 \ge 1$ and a quantum algorithm $\mathrm{QMC}_1(\mathcal{O}, \epsilon, \delta)$ which returns an estimate $\hat{y}$ of $\mathbb{E}[y]$ such that $P(|\hat{y} - \mathbb{E}[y]| \le \epsilon) \ge 1-\delta$ using at most $\frac{C_1}{\epsilon}\log\frac{1}{\delta}$ queries to $\mathcal{O}$ and $\mathcal{O}^\dagger$.*

Lemma 3 (Quantum multivariate mean estimator (Cornelissen et al., 2022))**.** *Assume that $\boldsymbol{y} : \Omega \to [0,1]^d$ is a d-dimensional random variable, Ω is equipped with a probability measure P, and the quantum unitary oracle $\mathcal{O}$ encodes P and $\boldsymbol{y}$. There is a constant $C_2 \ge 1$ and a quantum algorithm $\mathrm{QMC}_2(\mathcal{O}, \epsilon, \delta)$ which returns an estimate $\hat{\boldsymbol{y}}$ of $\mathbb{E}[\boldsymbol{y}]$ such that $P(\|\hat{\boldsymbol{y}} - \mathbb{E}[\boldsymbol{y}]\|_\infty \le \epsilon) \ge 1-\delta$ using at most $\frac{C_2\sqrt{d}\log(d/\delta)}{\epsilon}\sqrt{\log\left(\frac{\sqrt{d}\log(d/\delta)}{\epsilon}\right)}$ queries to $\mathcal{O}$ and $\mathcal{O}^\dagger$.*

3 PROBLEM-INDEPENDENT QUANTUM BWK

In this section, we develop a problem-independent quantum algorithm for BwK that improves the regret bound by jointly exploiting quantum reward estimation and adaptive resource control. The key challenge is that reward estimation can benefit from coherent quantum queries, whereas resource consumption must be tracked after every arm pull to respect the knapsack constraints. Our algorithm separates these two roles: it estimates rewards in batches using quantum mean estimation, while

updating resource consumption and the primal-dual weights after every pull. This design allows the tighter quantum confidence bounds to directly improve arm selection while maintaining fine-grained control of the remaining resources.

We present the pseudocode in Algorithm 1. The algorithm accesses quantum oracles for both reward and resource consumption as defined in Definition 1. Reward queries are processed coherently to construct quantum UCB estimates, while the resource consumption oracle is measured immediately after each pull. Accordingly, reward confidence bounds are updated after batched quantum queries, whereas resource confidence bounds are updated round by round.

During the execution, each arm is first played for N consecutive rounds to obtain initial confidence bounds for its reward and resource consumption. The algorithm then selects the arm maximizing the "bang-per-buck" ratio, i.e., the estimated reward relative to the estimated resource cost. After each pull, the resource bounds are updated immediately, while the reward bound is refreshed once the selected arm has accumulated sufficiently many additional quantum queries. We use $\boldsymbol{r}^U(s)$ and $\boldsymbol{C}^L(s)$ to denote the reward UCB and resource consumption LCB at stage s, respectively, and let τ denote the stopping time of Phase II.

Algorithm 1: Problem-Independent Quantum BwK

begin
  `// Phase I: Initialization`
  for $i = 1 \to m$ **do**
    $\mathrm{rad}_i \leftarrow 1$ and $N \leftarrow \frac{2C_1}{\mathrm{rad}_i} \log T$;
    Play arm i for consecutive N rounds ;
    Run QMC to get UCB $r_i^U(1) \in [0,1]$ of the reward and run classical algorithm to get LCB $\boldsymbol{C}_{\cdot,i}^L(1)$ for resource consumption;
    $\mathrm{rad}_i \leftarrow \mathrm{rad}_i/2$;
  $\boldsymbol{v}(1) = \mathbf{1}_d$;
  `// Phase II: pick arms maximize the "bang-per-buck" ratio`
  for *time* $s = 1, 2, \ldots, \tau$ **do**
    Let $i(s) \leftarrow \arg\max_i r_i^U(s)/(\boldsymbol{C}_{\cdot,i}^L(s) \cdot \boldsymbol{v}(s))$;
    Play arm $i(s)$ once and get $\boldsymbol{C}_{\cdot,i(s)}^L(s+1)$ classically ;
    if *arm* $i(s)$ *has been played for* $\frac{2C_1}{\mathrm{rad}_{i(s)}} \log T$ *times since the last execution of QMC* **then**
      Run QMC to update $r_i^U(s)$, $\mathrm{rad}_{i(s)} \leftarrow \mathrm{rad}_{i(s)}/2$;
    $v_j(s+1) \leftarrow v_j(s)(1+\epsilon)^{C_{j,i(s)}^L(s)}$, for all $j \in [d]$;

Theorem 1 (Main theorem, problem-independent setting)**.** *The expected regret of Algorithm 1 satisfies* $\mathrm{OPT}_{\mathrm{LP}} - \mathrm{REW} \le \mathcal{O}(\sqrt{\log(dT)})(\mathrm{OPT}_{\mathrm{LP}}\sqrt{\frac{m}{B}}) + \mathcal{O}(m \log(dT) \log T)$.

The proof of Theorem 1 is deferred to Appendix B. The quantum time complexity of this algorithm is dominated by the quantum Monte Carlo subroutine for reward estimation. Since an estimate with N oracle queries costs $\tilde{\mathcal{O}}(N)$ time, and the algorithm uses at most T oracle queries in total, the overall quantum time complexity is $\tilde{\mathcal{O}}(T)$.

4 Problem-Dependent Quantum BwK

In this section, we develop a problem-dependent quantum algorithm for BwK that improves both regret and time complexity. The algorithm combines quantum exploration for identifying optimal arms and non-binding constraints with adaptive LP-based allocation for exhausting binding resources. Its two-phase design exploits quantum speedups in both statistical estimation and LP optimization, while allowing coarse approximate LP solutions throughout and establishing robustness guarantees that lead to improved regret and computational complexity.

We present the pseudocode of our algorithm in Algorithm 2. At time step t, we denote by $\boldsymbol{r}^U(t)$ and $\boldsymbol{r}^L(t)$ the UCB and LCB of the reward vector, and by $\boldsymbol{C}^U(t)$ and $\boldsymbol{C}^L(t)$ the UCB and LCB of the resource consumption matrix, respectively. We let $\boldsymbol{B}^{(t)}$ denote the remaining resource budget

at time t. The algorithm comprises two phases. In Phase I, our objective is to identify all optimal arms $i \in \mathcal{I}^*$ and non-binding constraints $j \in \mathcal{J}'$. In each iteration of the while loop in Phase I, the algorithm plays all arms multiple times and queries the quantum reward and resource consumption oracles. Subsequently, the quantum Monte Carlo method is employed to compute the UCB and LCB for the reward and resource consumption of each arm.

Algorithm 2: Problem-Dependent Quantum BwK

begin
  // Phase I: Identification of $\mathcal{I}^*$ and $\mathcal{J}'$
  Initialize $\hat{\mathcal{I}}^* = \hat{\mathcal{J}}' = \emptyset$, $r = 1$, $t = 0$, $\boldsymbol{B}^{(0)} = \boldsymbol{B}$;
  $\theta = \min\left\{\frac{\min\{1,\sigma^2\}\min\{\chi,\delta\}}{12\min\{m^2,d^2\}}, \left(2+\frac{1}{b}\right)^{-2}\cdot\frac{\delta}{5}\right\}$, $\epsilon = \frac{\min\{1,\sigma\}\min\{\chi,\delta\}b}{5d^{3/2}}$;
  // The exact values of parameters are not required; suitable lower bounds suffice (see Appendix C).
  while $|\hat{\mathcal{I}}^*| + |\hat{\mathcal{J}}'| \le d$ **do**
    for $i = 1 \to m$ **do**
      $N \leftarrow \frac{\log T}{r}$, $r \leftarrow r/2$;
      Play arm i for N consecutive rounds;
      Run QMC to get UCB and LCB for the reward and resource consumption of arm i ;
      $t \leftarrow t + N$;
    Solve the LCB problem for $\mathrm{OPT}^L_{\mathrm{LP}}(t)$ in (10a).
    for $i \notin \hat{\mathcal{I}}^*$ **do**
      Solve the UCB problem for $\mathrm{OPT}^U_i(t)$ in (10b).
      if $\mathrm{OPT}^L_{\mathrm{LP}}(t) > \mathrm{OPT}^U_i(t)$ **then** Update $\hat{\mathcal{I}}^* = \hat{\mathcal{I}}^* \cup \{i\}$;
    for $j \notin \hat{\mathcal{J}}'$ **do**
      Solve the UCB problem for $\mathrm{OPT}^U_j(t)$ in (10c).
      if $\mathrm{OPT}^L_{\mathrm{LP}}(t) > \mathrm{OPT}^U_j(t)$ **then** Update $\hat{\mathcal{J}}' = \hat{\mathcal{J}}' \cup \{j\}$;
    Update $t = t + 1$ and $\boldsymbol{B}^{(t)}$;
  // Phase II: Exhausting the binding resources
  while $t \le \tau$ **do**
    if $\|\boldsymbol{C}^L_{\cdot,i}(t) - \boldsymbol{C}_{\cdot,i}\|_\infty > \theta$ *for some* $i \in \hat{\mathcal{I}}^*$ *or* $\boldsymbol{B}^{(t-1)}/(T-t+1) \notin [\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]$ **then**
      Solve the following LP

$$\max_{\substack{\boldsymbol{\xi}\ge\boldsymbol{0},\ \xi_i=0,\ i\notin\hat{\mathcal{I}}^* \\ \boldsymbol{C}^L(t-1)\boldsymbol{\xi}\le\boldsymbol{B}^{(t-1)}}} \left(\boldsymbol{r}^U(t-1)\right)^\top \boldsymbol{\xi}. \tag{8}$$

    else
      Solve the following LP

$$\max_{\substack{\boldsymbol{\xi}\ge\boldsymbol{0},\ \xi_i=0,\ i\notin\hat{\mathcal{I}}^* \\ \boldsymbol{C}^L(t-1)\boldsymbol{\xi}\le\boldsymbol{B}^{(t-1)},\ \boldsymbol{C}^L_{\hat{\mathcal{J}}^*,\hat{\mathcal{I}}^*}(t-1)\boldsymbol{\xi}_{\hat{\mathcal{I}}^*}\ge\boldsymbol{B}^{(t-1)}_{\hat{\mathcal{J}}^*}}} \left(\boldsymbol{r}^U(t-1)\right)^\top \boldsymbol{\xi}. \tag{9}$$

      Denote its optimal solution as $\tilde{\xi}$;
      Normalize $\tilde{\xi}$ into a probability and randomly play an arm according to the probability (using classical reward and resource consumption oracles) ;
      Update UCB and LCB of rewards and resources consumption by classical Monte Carlo ;
      Update $\boldsymbol{B}^{(t)}$ and $t = t + 1$;

To formalize the decision-making process, we define a sequence of LPs that are solved in this phase.

$$\mathrm{OPT}^L_{\mathrm{LP}}(t) := \max_{\boldsymbol{\xi}\ge\boldsymbol{0},\ \boldsymbol{C}^U(t)\boldsymbol{\xi}\le\boldsymbol{B}} \boldsymbol{r}^L(t)^\top\boldsymbol{\xi}, \tag{10a}$$

$$\mathrm{OPT}^U_i(t) := \max_{\boldsymbol{\xi}\ge\boldsymbol{0},\ \boldsymbol{C}^L(t)\boldsymbol{\xi}\le\boldsymbol{B},\ \xi_i=0} \boldsymbol{r}^U(t)^\top\boldsymbol{\xi}, \tag{10b}$$

$$\mathrm{OPT}^U_j(t) := \min_{\boldsymbol{\eta}\ge\boldsymbol{0},\ \boldsymbol{C}^L(t)^\top\boldsymbol{\eta}\ge\boldsymbol{r}^U(t)+\boldsymbol{C}^U_{j,\cdot}(t)} \boldsymbol{B}^\top\boldsymbol{\eta} - B. \tag{10c}$$

The value $\mathrm{OPT}^L_{\mathrm{LP}}(t)$ provides a conservative estimate of the achievable reward, computed with UCB estimates of resource consumption and LCB estimates of reward. For each arm i, $\mathrm{OPT}^U_i(t)$

evaluates whether excluding arm i degrades the optimal value. For each constraint j, $\mathrm{OPT}_j^U(t)$ gives the corresponding dual LP for identifying non-binding constraints. As we know in Proposition 1, there exists a gap in the optimal value between $\mathrm{OPT}_{\mathrm{LP}}$ and OPT_i (or OPT_j). Provided that the expectations of $\boldsymbol{r}$ and $\boldsymbol{C}$ lie within the confidence interval, the following conditions hold: $\mathrm{OPT}_{\mathrm{LP}}^L(t) \leq \mathrm{OPT}_{\mathrm{LP}}$, $\mathrm{OPT}_i^U(t) \geq \mathrm{OPT}_i$, and $\mathrm{OPT}_j^U(t) \geq \mathrm{OPT}_j$. Consequently, by solving the UCB and LCB LP problems defined in (10b) and (10c) for each arm i and resource j, and by comparing these results with (10a), we can identify the optimal arms $i \in \mathcal{I}^*$ and non-binding constraints $j \in \mathcal{J}'$ with high probability.

Phase II of Algorithm 2 leverages the optimal arms $i \in \mathcal{I}^*$ identified in Phase I. In each round of this phase, an adaptive LP approach (8) is employed to efficiently utilize the remaining resources constrained by binding conditions. When the additional conditions on $\boldsymbol{C}_{\cdot,i}^L(t)$ and $\boldsymbol{B}^{(t-1)}$ are satisfied—namely, when the estimated resource consumption is accurate and the average remaining budget lies within the prescribed interval—the LP in (8) becomes equivalent to (9). In this regime, the additional binding-resource constraints provide two-sided control of the resource residuals under approximate feasibility, allowing us to bound the perturbation of the normalized arm-allocation probabilities. In both formulations, the decision variables ξ_i for arms $i \notin \mathcal{I}^*$ are fixed at zero, and the available capacity is updated to $\boldsymbol{B}^{(t-1)}$ rather than remaining at its initial value $\boldsymbol{B}$. Once the LP is solved, the optimal solution of this LP is then normalized into a probability distribution, according to which an arm is randomly selected in each round. We use the quantum oracles for reward and resource consumption by measuring after each oracle call to fully utilize the remaining resources.

Time complexity analysis To analyze the computational complexity of these LP subproblems, we formalize the approximation guarantee and oracle access required by the LP solver.

Definition 2 (ϵ_{LP}-approximate quantum LP solver)**.** *Consider a linear programming (LP) problem in the form of (1a):* $\max_{\boldsymbol{\xi} \in \mathbb{R}^n,\, \boldsymbol{\xi} \geq 0} \boldsymbol{r}^\top \boldsymbol{\xi},\ \text{s.t. } \boldsymbol{C}\boldsymbol{\xi} \leq \boldsymbol{B}$*, where all coefficients are bounded within the range* $[-1, 1]$*. Let* m *denote the number of variables and* d *the number of constraints. We define OPT as the optimal value of the LP,* $\boldsymbol{\xi}$ *as the optimal primal solution, and* $\boldsymbol{\eta}$ *as the optimal dual solution. Assume that* $\sum_i \xi_i \leq R$ *and* $\sum_i \eta_i \leq r$*. Given quantum oracles* $\mathcal{O}_{\boldsymbol{r}}, \mathcal{O}_{\boldsymbol{C}}$*, and* $\mathcal{O}_{\boldsymbol{B}}$ *that encode the coefficient matrix of the LP problem such that,*

$$\begin{aligned} \mathcal{O}_{\boldsymbol{r}} |i\rangle |0\rangle &= |i\rangle |r_i\rangle \text{, for } i \in [m] \\ \mathcal{O}_{\boldsymbol{C}} |i\rangle |j\rangle |0\rangle &= |i\rangle |j\rangle |C_{i,j}\rangle \text{, for } i \in [d],\, j \in [m], \\ \mathcal{O}_{\boldsymbol{B}} |i\rangle |0\rangle &= |i\rangle |B_i\rangle \text{, for } i \in [d]. \end{aligned}$$

A quantum LP solver is supposed to compute an ϵ_{LP}*-optimal and* ϵ_{LP}*-feasible solution* $\boldsymbol{\xi}^* \geq 0$*, satisfying* $|\boldsymbol{r}^\top \boldsymbol{\xi}^* - \mathrm{OPT}| \leq \epsilon_{\mathrm{LP}}$ *and* $\boldsymbol{C}\boldsymbol{\xi}^* \leq \boldsymbol{B} + \epsilon_{\mathrm{LP}}$*.*

The main computational cost of Algorithm 2 comes from constructing the coefficient oracles for the LP problems and solving these LPs. Since the LP coefficients depend on the UCB and LCB estimates of rewards and resource consumptions, the corresponding oracles need to be updated as new observations are collected. In Phase I, each update changes the confidence estimates for all arms and hence the coefficients of the identification LPs. In Phase II, the estimates and the residual LP are updated after every arm pull. To solve these LP problems, we employ the quantum LP solver in Lemma 4. Detailed discussion of quantum and classical LP solvers is provided in Appendix A.2.

Lemma 4 (Quantum LP solver (Gao et al., 2024))**.** *Under the setting of Definition 2, there exists a quantum algorithm that solves LP problems with time complexity* $\tilde{\mathcal{O}}((Rr)^{2.5}\sqrt{d+m}/\epsilon_{\mathrm{LP}}^{2.5})$*.*

We next characterize the approximation accuracy required for the LP subroutines in Algorithm 2 and the corresponding quantum time complexity.

Proposition 2 (Quantum complexity of approximate LP subroutines)**.** *Using the quantum LP solver in Lemma 4, the LP problems in (10a),(10b), and (10c) can be solved to* $\epsilon_{\mathrm{LP}} \cdot T$*-optimal and* $\epsilon_{\mathrm{LP}} \cdot T$*-approximate in quantum time complexity* $\tilde{\mathcal{O}}\left(\sqrt{m+d}/\epsilon_{\mathrm{LP}}^{2.5}\right)$*. The LPs in (8) and (9) can be solved to* $\epsilon_{\mathrm{LP}} \cdot (T-t)/\log^2 T$*-optimal and* $\epsilon_{\mathrm{LP}} \cdot (T-t)/\log^2 T$*-approximate in quantum time complexity* $\tilde{\mathcal{O}}\left(\sqrt{m+d}\log^5 T/\epsilon_{\mathrm{LP}}^{2.5}\right)$*.*

A key contribution of Proposition 2 is that Algorithm 2 requires only coarse approximate LP solutions in both phases. The admissible additive errors scale as $\epsilon_{\mathrm{LP}} T$ in Phase I and $\epsilon_{\mathrm{LP}}(T-t)/\log^2 T$

in Phase II, rather than requiring a fixed absolute precision or near-exact solutions as the horizon grows. These accuracies are sufficient to resolve the separation gap in Phase I and to control the induced arm-allocation probabilities in Phase II. This property allows our algorithm to exploit fast approximate LP solvers throughout.

Theorem 2 (Time complexity)**.** *The overall time complexity of Algorithm 2 with the quantum LP solver in Lemma 4 is* $\tilde{\mathcal{O}}(md + (m+d)^{1.5}/\epsilon_{\mathrm{LP}}^{2.5} + (\sqrt{m+d}/\epsilon_{\mathrm{LP}}^{2.5} + d)T)$.

The proofs of Proposition 2 and Theorem 2 are deferred to Appendix C.1 and Appendix C.2.

Remark 1. *The ability to use approximate LP solutions is a property of our algorithm rather than of the particular quantum LP solver. For example, replacing Lemma 4 with the classical approximate LP solver based on zero-sum games (Grigoriadis & Khachiyan, 1995) gives the overall time complexity* $\widetilde{\mathcal{O}}(md + (m+d)^2/\epsilon_{\mathrm{LP}}^2 + ((m+d)/\epsilon_{\mathrm{LP}}^2 + d)T)$. *Our algorithm can therefore also exploit fast classical approximate LP solvers. In comparison, the classical problem-dependent algorithm of Li et al. (2021) requires its LP subproblems to be solved to near-exact precision; using the high-accuracy LP solver of Jiang et al. (2020) gives an overall complexity of* $\widetilde{\mathcal{O}}(\max\{m,d\}^{2.372}(T+m+d))$. *Allowing coarse approximate LP solutions is therefore a key contribution for the improved dependence of our overall time complexity on the problem dimensions.*

Regret analysis and robustness of solving LP problems approximately The regret of Algorithm 2 comes from two parts: pulling sub-optimal arms during identification and leaving binding resources unused at the end of the adaptive allocation. Phase I reduces the first term through quantum exploration, while Phase II controls the second term through adaptive allocation based on coarse approximate LP solutions. We analyze the two phases separately below.

In Phase I, the algorithm identifies the optimal arms $\mathcal{I}^*$ and non-binding constraints $\mathcal{J}'$ by comparing the confidence-bound LP values $\mathrm{OPT}_{\mathrm{LP}}^L(t)$, $\mathrm{OPT}_i^U(t)$, and $\mathrm{OPT}_j^U(t)$. As the confidence intervals shrink, the LP values corresponding to $i \in \mathcal{I}^*$ and $j \in \mathcal{J}'$ become separated from $\mathrm{OPT}_{\mathrm{LP}}^L(t)$. The additive error introduced by the approximate LP solver can be absorbed by the problem-dependent separation gap δ. Quantum mean estimation provides tighter UCB and LCB estimates, reducing the number of pulls required and leading to the following regret bound.

Proposition 3 (Expected regret in Phase I)**.** *With* $\epsilon_{\mathrm{LP}} \le \frac{\delta}{4}$, *the expected regret in Phase I of Algorithm 2 is upper bounded by* $\tilde{\mathcal{O}}((2+\frac{1}{b})\frac{m\sqrt{d}\log T}{b\delta})$.

In Phase II, the regret is determined by the amount of unused binding resources $\boldsymbol{B}_{\mathcal{J}^*}^{(\tau)}$ at termination. The main difficulty is that the residual LP solution is normalized into an arm-pulling distribution, so LP approximation errors directly perturb the allocation probabilities and, consequently, the evolution of the remaining resource budget.

Proposition 4 (Expected regret in Phase II)**.** *With* $\epsilon_{\mathrm{LP}} \le \min\{\frac{\sigma\chi\log^2 T}{40s^{3/2}}, \frac{2b\theta\min\{\chi,\delta\}\log^2 T}{405}\}$, *the expected regret in Phase II of Algorithm 2 is upper bounded by* $\mathcal{O}\left(\frac{d^4}{b^2\min\{\chi^2,\delta^2\}\min\{1,\sigma^2\}}\right)$.

The core of our proof hinges on the adaptive strategy of the algorithm. We formulate two LP problems, and (9) incorporates an additional constraint compared to (8). We demonstrate that if the resource consumption matrix is accurately estimated, such that $\|\boldsymbol{C}_{\cdot,i}^L(t) - \boldsymbol{C}_{\cdot,i}\|_\infty \le \theta$ for all $i \in \mathcal{I}^*$, and the average remaining resource budget is appropriately bounded, i.e., $\boldsymbol{B}^{(t-1)}/(T-t+1) \in [\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]$, then the normalized approximate solution is bounded away from zero on each optimal arm. This lower bound ensures that every optimal arm is sampled sufficiently often so that the binding resources are consumed at comparable rates, leaving only a problem-dependent constant amount unused in expectation at termination.

The proofs of Proposition 3 and Proposition 4 are deferred to Appendix C.3 and Appendix C.4. By combining the regret bounds in Phase I and Phase II, we obtain the expected regret upper bound Theorem 3 for our quantum algorithm.

Theorem 3 (Main theorem, problem-dependent setting)**.** *Under Assumption 1, with problem-dependent parameters* δ, χ, σ *and* s *defined in (6), the expected regret of Algorithm 2 is upper bounded by* $\tilde{\mathcal{O}}((2+\frac{1}{b})\frac{m\sqrt{d}}{b\delta}\log T + \frac{d^4}{b^2\min\{\chi^2,\delta^2\}\min\{1,\sigma^2\}})$ *provided that* $\epsilon_{\mathrm{LP}} \le \min\{\frac{\sigma\chi\log^2 T}{40s^{3/2}}, \frac{2b\theta\min\{\chi,\delta\}\log^2 T}{405}, \frac{\delta}{4}\}$.

AI USE STATEMENT

Generative AI tools were used only to aid and polish the writing of this paper. They were not used for any research tasks requiring disclosure under the ICLR 2027 policy. All AI-assisted text was reviewed by the authors, who take responsibility for the final content.

REFERENCES

Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bärtschi, Harry Buhrman, Carleton Coffrin, Giorgio Cortiana, Vedran Dunjko, Daniel J. Egger, Bruce G. Elmegreen, Nicola Franco, Filippo Fratini, Bryce Fuller, Julien Gacon, Constantin Gonciulea, and Christa Zoufal. Challenges and opportunities in quantum optimization. *Nature Reviews Physics*, 6(12):718–735, December 2024. ISSN 2522-5820. doi: 10.1038/s42254-024-00770-9. URL https://doi.org/10.1038/s42254-024-00770-9.

Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014. arXiv:0911.2974

Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2005–2039. SIAM, 2025. arXiv:2404.16349

Joran van Apeldoorn and András Gilyén. Quantum algorithms for zero-sum games. *arXiv preprint arXiv:1904.03180*, 2019a. arXiv:1904.03180

Joran van Apeldoorn and András Gilyén. Improvements in quantum SDP-solving with applications. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 99:1–99:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019b. arXiv:1804.05058

Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum SDP-solvers: Better upper and lower bounds. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 403–414. IEEE, 2017. arXiv:1705.01843

Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Convex optimization using quantum oracles. *Quantum*, 4:220, 2020. arXiv:1809.00643

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Brandon Augustino, Jiaqi Leng, Giacomo Nannicini, Tamás Terlaky, and Xiaodi Wu. A quantum central path algorithm for linear optimization. *arXiv preprint arXiv:2311.03977*, 2023a. arXiv:2311.03977

Brandon Augustino, Giacomo Nannicini, Tamás Terlaky, and Luis F. Zuluaga. Quantum interior point methods for semidefinite optimization. *Quantum*, 7:1110, 2023b. arXiv:2112.06025

Brandon Augustino, Dylan Herman, Enrico Fontana, Junhyung Lyle Kim, Jacob Watkins, Shouvanik Chakrabarti, and Marco Pistoia. Fast convex optimization with quantum gradient methods. *arXiv preprint arXiv:2503.17356*, 2025. arXiv:2503.17356

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *Journal of the ACM (JACM)*, 65(3):1–55, 2018. arXiv:1305.2545

Dimitris Bertsimas and José Niño-Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Operations Research*, 48(1):80–90, 2000.

Omar Besbes and Assaf Zeevi. Blind network revenue management. *Operations Research*, 60(6): 1537–1550, 2012.

Jose Blanchet, Mario Szegedy, and Guanyang Wang. Quadratic speed-up in infinite variance quantum Monte Carlo. *arXiv preprint*, 2024. arXiv:2401.07497

Jose Blanchet, Yassine Hamoudi, Mario Szegedy, and Guanyang Wang. Quantum speedup of nonlinear Monte Carlo problems. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. arXiv:2502.05094

Vivek S. Borkar and Sean P. Meyn. Risk-sensitive optimal control for Markov decision processes with monotone cost. *Mathematics of Operations Research*, 27(1):192–209, 2002.

Adam Bouland, Yosheb M. Getachew, Yujia Jin, Aaron Sidford, and Kevin Tian. Quantum speedups for zero-sum games via improved dynamic Gibbs sampling. In *International Conference on Machine Learning*, pp. 2932–2952. PMLR, 2023. arXiv:2301.03763

Stephen P. Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Jan van den Brand, Yin Tat Lee, Yang P. Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, MDPs, and ℓ_1-regression in nearly linear time for dense instances. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 859–869, 2021. arXiv:2101.05719

Fernando G. S. L. Brandão, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu. Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi (eds.), *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 27:1–27:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-109-2. doi: 10.4230/LIPIcs.ICALP.2019.27. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2019.27. arXiv:1710.02581

Fernando G.S.L. Brandão and Krysta Svore. Quantum speed-ups for semidefinite programming. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science*, pp. 415–426, 2017. arXiv:1609.05537

Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. arXiv:quant-ph/0005055

Shouvanik Chakrabarti, Andrew M. Childs, Tongyang Li, and Xiaodi Wu. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4:221, 2020. arXiv:1809.01731

Zherui Chen, Yuchen Lu, Hao Wang, Yizhou Liu, and Tongyang Li. Quantum Langevin dynamics for optimization. *Communications in Mathematical Physics*, 406(3):52, 2025. arXiv:2311.15587

Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM (JACM)*, 68(1):1–39, 2021. arXiv:1810.07896

Arjan Cornelissen, Yassine Hamoudi, and Sofiene Jerbi. Near-optimal quantum algorithms for multivariate mean estimation. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 33–43, 2022. arXiv:2111.09787

Zhongxiang Dai, Gregory Kang Ruey Lau, Arun Verma, Yao Shu, Bryan Kian Hsiang Low, and Patrick Jaillet. Quantum Bayesian optimization. *Advances in Neural Information Processing Systems*, 36:20179–20207, 2023. arXiv:2310.05373

George B Dantzig. Linear programming and extensions. 2016.

Vedran Dunjko and Hans J. Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, 2018.

Kris Johnson Ferreira, David Simchi-Levi, and He Wang. Online network revenue management using thompson sampling. *Operations Research*, 66(6):1586–1602, 2018.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Minbo Gao, Zhengfeng Ji, Tongyang Li, and Qisheng Wang. Logarithmic-regret quantum learning algorithms for zero-sum games. *Advances in Neural Information Processing Systems*, 36, 2024. arXiv:2304.14197

Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical Review Letters*, 100(16):160501, 2008. arXiv:0708.1879

John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

Weiyuan Gong, Chenyi Zhang, and Tongyang Li. Robustness of quantum algorithms for nonconvex optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. arXiv:2212.02548

Michael D. Grigoriadis and Leonid G. Khachiyan. A sublinear-time randomized approximation algorithm for matrix games. *Operations Research Letters*, 18(2):53–58, 1995.

Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002. arXiv:quant-ph/0208112

Sofiene Jerbi, Arjan Cornelissen, Māris Ozols, and Vedran Dunjko. Quantum policy gradient algorithms. *arXiv preprint arXiv:2212.09328*, 2022. arXiv:2212.09328

Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. Faster dynamic matrix inverse for faster LPs. *arXiv preprint arXiv:2004.07470*, 2020. arXiv:2004.07470

Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pp. 302–311, 1984.

Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, pp. 49–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017. arXiv:1603.08675

Leonid Genrikhovich Khachiyan. A polynomial algorithm in linear programming. In *Doklady Akademii Nauk*, volume 244, pp. 1093–1096. Russian Academy of Sciences, 1979.

Robin Kothari and Ryan O'Donnell. Mean estimation when you have the source code; or, quantum Monte Carlo methods. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1186–1215. SIAM, 2023. arXiv:2208.07544

Francois Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the Coppersmith-Winograd tensor. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1029–1046. SIAM, 2018.

Yin Tat Lee and Aaron Sidford. Efficient inverse maintenance and faster algorithms for linear programming. In *2015 IEEE 56th annual symposium on foundations of computer science*, pp. 230–249. IEEE, 2015. arXiv:1503.01752

Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 1049–1065. IEEE, 2015. arXiv:1508.04874

Jiaqi Leng, Ethan Hickman, Joseph Li, and Xiaodi Wu. Quantum Hamiltonian descent. *arXiv preprint arXiv:2303.01471*, 2023. arXiv:2303.01471

Jiaqi Leng, Kewen Wu, Xiaodi Wu, and Yufan Zheng. (sub) exponential quantum speedup for optimization. *arXiv preprint arXiv:2504.14841*, 2025. arXiv:2504.14841

Xiaocheng Li, Chunlin Sun, and Yinyu Ye. The symmetry between arms and knapsacks: A primal-dual approach for bandits with knapsacks. In *International Conference on Machine Learning*, pp. 6483–6492. PMLR, 2021. arXiv:2102.06385

Yizhou Liu, Weijie J. Su, and Tongyang Li. On quantum speedups for nonconvex optimization via quantum tunneling walks. *Quantum*, 7:1030, 2023. arXiv:2209.14501

Nimrod Megiddo and Ramaswamy Chandrasekaran. On the ε-perturbation method for avoiding degeneracy. *Operations Research Letters*, 8(6):305–308, 1989.

Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22–es, 2007.

Nico Meyer, Christian Ufrecht, Maniraman Periyasamy, Daniel D. Scherer, Axel Plinge, and Christopher Mutschler. A survey on quantum reinforcement learning. *arXiv preprint arXiv:2211.03464*, 2022. arXiv:2211.03464

Ashley Montanaro. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, 2015. arXiv:1504.06987

Giacomo Nannicini. Fast quantum subroutines for the simplex method. *Operations Research*, 72 (2):763–780, 2024. arXiv:1910.10649

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

Andrzej Ruszczyński. Risk-averse dynamic programming for Markov decision processes. *Mathematical programming*, 125:235–261, 2010.

Zakaria Shams Siam, Chaowen Guan, and Chong Liu. Quantum non-linear bandit optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25454–25462, 2026. arXiv:2503.03023

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019. arXiv:1904.07272

Zongqi Wan, Zhijie Zhang, Tongyang Li, Jialin Zhang, and Xiaoming Sun. Quantum multi-armed bandits and stochastic linear bandits enjoy logarithmic regrets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10087–10094, 2023. arXiv:2205.14988

Daochen Wang, Xuchen You, Tongyang Li, and Andrew M. Childs. Quantum exploration algorithms for multi-armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10102–10110, 2021. arXiv:2007.07049

Mengdi Wang. Randomized linear programming solves the Markov decision problem in nearly linear (sometimes sublinear) time. *Mathematics of Operations Research*, 45(2):517–546, 2020.

Richard Weber. On the Gittins index for multiarmed bandits. *The Annals of Applied Probability*, pp. 1024–1033, 1992.

Yulian Wu, Chaowen Guan, Vaneet Aggarwal, and Di Wang. Quantum heavy-tailed bandits. *arXiv preprint*, 2023. arXiv:2301.09680

Jia Yuan Yu, Shie Mannor, and Nahum Shimkin. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757, 2009.

Chenyi Zhang and Tongyang Li. Quantum lower bounds for finding stationary points of nonconvex functions. In *International Conference on Machine Learning*, pp. 41268–41299. PMLR, 2023. arXiv:2212.03906

Han Zhong, Jiachen Hu, Yecheng Xue, Tongyang Li, and Liwei Wang. Provably efficient exploration in quantum reinforcement learning with logarithmic worst-case regret. *arXiv preprint arXiv:2302.10796*, 2023. arXiv:2302.10796

A Additional Preliminaries

A.1 Quantum oracle models and quantum Monte Carlo

In quantum computing, a single qubit is represented by a vector in the complex Hilbert space $\mathbb{C}^2$, while an n-qubit quantum state resides in the space $\mathbb{C}^{2^n}$. Given two quantum states $|v\rangle \in \mathbb{C}^{2^{n_1}}$ and $|u\rangle \in \mathbb{C}^{2^{n_2}}$, their tensor product $|v\rangle \otimes |u\rangle$ is a vector in $\mathbb{C}^{2^{n_1+n_2}}$ of the form $(v_0u_0, v_0u_1, \ldots, v_{2^{n_1}-1}u_{2^{n_2}-1})$. For brevity, this tensor product is often denoted by $|v\rangle\,|u\rangle$.

The quantum oracles in Definition 1 prepare a coherent superposition over all possible outcomes from the underlying reward and cost distributions associated with each arm. This definition naturally generalizes the classical sampling oracle: if a measurement is performed on the state $\mathcal{O}_i^r\,|0\rangle$ or $\mathcal{O}_i^C\,|0\rangle$, the quantum oracle behaves like a classical oracle by returning a single sample according to the corresponding distribution. In contrast to classical oracles, which provide the exact reward and resource costs associated with each arm pull, quantum oracles provide no direct feedback until a measurement is performed. When used within a quantum algorithm, the algorithm can query both the unitary oracle $\mathcal{O}$ and its inverse $\mathcal{O}^\dagger$ during each interaction with the arm.

In addition to the sampling oracles for rewards and resource consumption, the quantum LP solver used in this paper requires oracle access to the coefficients of the linear programming problem (see Definition 2 for details). We implement this oracle by storing the coefficients in a quantum-read, classical-write random access memory (QRAM) (Giovannetti et al., 2008). Specifically, QRAM allows for the storage and modification of an array $a_1, a_2, \ldots, a_l$ of classical data while providing quantum read-only access. That is, it enables the implementation of a unitary operator $\mathcal{U}_{\text{QRAM}}$ such that

$$\mathcal{U}_{\text{QRAM}}\colon\ |i\rangle\,|0\rangle \to |i\rangle\,|a_i\rangle\,. \tag{11}$$

Given l classical data points, the QRAM can be constructed in $\tilde{\mathcal{O}}(l)$ time and allows access to each element in $\tilde{\mathcal{O}}(1)$ time (Kerenidis & Prakash, 2017; Grover & Rudolph, 2002).

We next introduce the basic idea behind quantum Monte Carlo methods; see also recent related works on nonlinear and infinite-variance Monte Carlo problems (Blanchet et al., 2024; 2026). In the quantum setting, QMC uses the coherent oracle access described above to estimate expectations with a nearly quadratic improvement over classical Monte Carlo. This acceleration stems from quantum amplitude estimation (Brassard et al., 2002), as detailed below. Suppose we have a quantum algorithm $\mathcal{A}$ acting on n qubits, whose output state is

$$|\phi\rangle := \mathcal{A}\,|0\rangle^{\otimes n} = \sum_x \sqrt{p(x)}\,|x\rangle\,, \tag{12}$$

where $\{|x\rangle \mid x \in \{0,1\}^n\}$ denotes the set of 2^n computational basis states, and $p(x) \in [0,1]$ is the probability of obtaining outcome x upon measurement. If the measured outcome is x, the algorithm returns a value $\phi(x)$, and our goal is to estimate the expectation $\mathbb{E}[\phi] = \sum_x p(x)\phi(x)$. To achieve this, the QMC employs quantum amplitude estimation. Specifically, we introduce one ancillary qubit and apply a unitary operator W defined by

$$|x\rangle\,|0\rangle \to |x\rangle\,(\sqrt{\phi(x)}\,|1\rangle + \sqrt{1-\phi(x)}\,|0\rangle). \tag{13}$$

Consequently, applying $\mathcal{A}$ on the first n qubits followed by W on the appended qubit transforms the initial state $|0\rangle^{\otimes(n+1)}$ into

$$\begin{aligned} |\psi\rangle := W(\mathcal{A}\otimes I)\,|0\rangle^{\otimes(n+1)} &= \sum_x \left(\sqrt{p(x)\phi(x)}\,|x\rangle\,|1\rangle + \sqrt{p(x)(1-\phi(x))}\,|x\rangle\,|0\rangle\right) \\ &= \sqrt{\mathbb{E}[\phi]}\,|\psi_1\rangle\,|1\rangle + \sqrt{1-\mathbb{E}[\phi]}\,|\psi_0\rangle\,|0\rangle\,, \end{aligned} \tag{14}$$

where $|\psi_1\rangle := \frac{1}{\sqrt{\mathbb{E}[\phi]}}\sum_x \sqrt{p(x)\phi(x)}\,|x\rangle$ and $|\psi_0\rangle := \frac{1}{\sqrt{1-\mathbb{E}[\phi]}}\sum_x \sqrt{p(x)(1-\phi(x))}\,|x\rangle$. In this form, the probability of measuring the ancilla in $|1\rangle$ is exactly $\mathbb{E}[\phi]$. Amplitude estimation then proceeds by alternating reflections about $|\psi\rangle$ and about the bad subspace $|\psi_0\rangle\,|0\rangle$ through the reflection on the last qubit. After t such iterations, one obtains an estimate $\tilde{\phi}$ satisfying $|\tilde{\phi} - \mathbb{E}[\phi]| \le \mathcal{O}(\sqrt{\mathbb{E}[\phi]}/t + 1/t^2)$ (Brassard et al., 2002). This leads to the fast quantum univariate and multivariate mean estimators stated in Lemma 2 and Lemma 3.

A.2 Quantum and classical algorithms for linear programming

Linear programming (LP) is a core topic in convex optimization (Boyd & Vandenberghe, 2004). Among methods for solving LP problems, the simplex method introduced by Dantzig (Dantzig, 2016) solves LP problems by traversing the edges of the feasible region and presents efficiency in practice. However, in the worst case, the simplex method may take an exponential number of iterations with respect to the problem size. In terms of polynomial-time algorithms,Khachiyan (1979) was the first to prove that LP problems can be solved in polynomial time using the ellipsoid method, thereby establishing that LP belongs to the complexity class P. The same conclusion was also established byKarmarkar (1984) using the interior-point method. At the moment, the state-of-the-art time complexities of classical LP solvers depend on the specific structure of the problem, such as the relationship between the number of variables m and constraints d, or the sparsity of the constraint matrix. When $d = \Omega(m)$, Cohen et al. (2021) and Jiang et al. (2020) solve the LP in current matrix multiplication time $\tilde{\mathcal{O}}(m^\omega)$. Another key result is the $\tilde{\mathcal{O}}(md + d^{2.5})$ complexity fromBrand et al. (2021), which is better when d is small. For sparse LP problems,Lee & Sidford (2015) proposed an algorithm with time complexity $\tilde{\mathcal{O}}((\mathrm{nnz}(\boldsymbol{A}) + m^2)\sqrt{m})$, where $\mathrm{nnz}(\boldsymbol{A})$ is the number of nonzero elements in $\boldsymbol{A}$.

While the aforementioned classical LP solvers have a poly-logarithmic dependence on the inverse of the desired precision, approximate LP solvers are also available. Specifically, an LP problem can be reduced to a zero-sum matrix game. Without loss of generality, assume that all coefficients in the LP are within the range $[-1, 1]$. Denote the optimal value of the LP as OPT, the optimal primal solution as $\boldsymbol{\xi}$, and the optimal dual solution as $\boldsymbol{\eta}$. Assume that $\sum_i \xi_i \leq R$ and $\sum_i \eta_i \leq r$. In this case, it follows that the optimal value, OPT, is bounded by R, i.e., $-R \leq \text{OPT} \leq R$. The reduction process involves transforming the LP problem into a decision problem: determining whether OPT satisfies OPT $> \alpha$ or OPT $\leq \alpha + \epsilon$. By performing binary search over the range $[-R, R]$, the LP problem can be solved in $\log(R/\epsilon)$ iterations. Furthermore, the decision problem per se can be reduced to computing an ϵ-approximate Nash equilibrium of the corresponding zero-sum game. In particular, Apeldoorn & Gilyén (2019a, Lemma 12) shows that solving the zero-sum games up to additive error $\epsilon'' = \epsilon/(6R(r+1))$ suffices to correctly conclude the decision problem. SinceGrigoriadis & Khachiyan (1995) proposed a classical algorithm that finds an ϵ-approximate Nash equilibrium of an $m \times d$ zero-sum game with time complexity $\tilde{\mathcal{O}}((m+d)/\epsilon^2)$, it implies a clssical algorithm for finding an ϵ-optimal and ϵ-feasible solution of an LP problem with time complexity $\tilde{\mathcal{O}}((m+d)(R(r+1)/\epsilon)^2)$.

On the quantum side, existing approaches for solving LPs are approximate and primarily offer polynomial improvements in complexity with respect to the problem dimensions. The best-known quantum algorithm for computing an ϵ-approximate Nash equilibrium in zero-sum games utilizes a faster Gibbs sampler and achieves a time complexity of $\tilde{\mathcal{O}}(\sqrt{m+n}/\epsilon^{2.5})$ (Bouland et al., 2023; Gao et al., 2024). By leveraging the reduction from LP to zero-sum games, this leads to the development of a quantum LP solver with improved time complexity of $\tilde{\mathcal{O}}(\sqrt{m+d}(R(r+1)/\epsilon)^{2.5})$. There also exist other quantum LP solvers with different complexity forms. For instance, Augustino et al. (2023a) proposed a quantum central path algorithm for LPs with complexity $\tilde{\mathcal{O}}(\sqrt{m+d} \cdot \mathrm{nnz}(\boldsymbol{A})R/\epsilon)$. More recently, Augustino et al. (2025) proposed a quantum LP solver based on a quantum version of the mirror descent method with time complexity $\tilde{\mathcal{O}}(d\sqrt{m}(R(r+1)/\epsilon)^2)$.

B Proofs for Problem-Independent Quantum BwK

In Algorithm 1 for problem-independent quantum BwK, both quantum and classical estimators are utilized to construct the UCB and LCB for reward and resource consumption. To analyze the algorithm and derive the upper bound of regret, we first present the following propositions, which demonstrate that, with high probability, the expected reward and resource consumption lie within the confidence intervals of the estimator. We begin by deriving the number of times the QMC has been invoked during the execution of the algorithm. Although similar proofs have been presented in Wan et al. (2023), we provide a self-contained proof of Lemma 5 here.

Lemma 5. *Let $\delta = \frac{1}{T^2}$ and C_1 be the constant in Lemma 2. The number of times that the quantum Monte Carlo method is invoked is at most*

$$m \log \left(\frac{T}{2mC_1 \log T} + 1 \right). \tag{15}$$

Proof. Given that there are T total rounds, in Phase I each arm is pulled $2C_1 \log T$ times, and the QMC is invoked m times. In Phase II, suppose each arm invokes the QMC N_i times. Summing the number of times each arm is played across the two phases, we obtain the inequality $\sum_{i=1}^{m} \left(2^{N_i+1} - 1\right) 2C_1 \log T \leq T$. Next, due to the convexity of the exponent function, $\sum_{i=1}^{m} 2^{N_i+1} \geq m 2^{1/m \sum_i N_i + 1}$. Therefore, we have

$$\sum_{i=1}^{m} (N_i + 1) \leq m \log \left(\frac{T}{2mC_1 \log T} + 1 \right) \leq m \log T, \tag{16}$$

which proves the lemma. □

Thus, the QMC method for estimating the reward has been invoked at most $m \log T$ times. With $\delta = \frac{1}{T^2}$ in Lemma 2, the probability that the expected reward lies within each confidence interval is at least $1 - \frac{m \log T}{T^2}$. Furthermore, the classical estimator for the d-dimensional resource consumption has been invoked at most T times. With $\delta = \frac{2d}{T^3}$ in Lemma 1, the expected resource consumption lies within the respective confidence interval with probability at least $1 - \frac{2d}{T^2}$. Consequently, with probability $1 - \frac{m \log T + 2d}{T^2}$, both the expected reward and resource consumption lie within their respective confidence bounds, thereby proving the following proposition.

Proposition 5. *Let $\hat{r}_i(s)$ denote the most recent output from QMC executions up to time s, and let $n_i^{\mathrm{QMC}}(s)$ represent the number of quantum oracle queries utilized in the QMC process. For each arm $i \in [m]$ at time $s \in [T]$, we define the following upper and lower confidence bounds for the reward:*

$$r_i^U(s) = \mathrm{proj}_{[0,1]} \left(\hat{r}_i(s) + \frac{2C_1 \log T}{n_i^{\mathrm{QMC}}(s)} \right), \quad r_i^L(s) = \mathrm{proj}_{[0,1]} \left(\hat{r}_i(s) - \frac{2C_1 \log T}{n_i^{\mathrm{QMC}}(s)} \right), \tag{17}$$

where $\mathrm{proj}_{[0,1]}$ denotes the projection onto the interval $[0, 1]$. Let $\hat{C}_{j,i}(s)$ denote the empirical mean of resource j observations, and let $n_i(s)$ denote the number of times arm i has been selected by time s. We define the following upper and lower bounds for the resource consumption:

$$C_{j,i}^U(s) = \hat{C}_{j,i}(s) + \sqrt{\frac{3 \log T}{n_i(s)}}, \quad C_{j,i}^L(s) = \hat{C}_{j,i}(s) - \sqrt{\frac{3 \log T}{n_i(s)}}. \tag{18}$$

Then, with probability at least $1 - \frac{m \log T + 2d}{T^2}$, the expected reward and resource consumption are contained within their respective confidence intervals:

$$r_i \in [r_i^L(s), r_i^U(s)],\ i \in [m],\ s \in [T], \quad C_{j,i} \in [C_{j,i}^L(s), C_{j,i}^U(s)],\ i \in [m],\ j \in [d],\ s \in [T] \tag{19}$$

Proof of Theorem 1 Once we have established the high-probability confidence intervals in Proposition 5, we essentially follow the proof outline ofBadanidiyuru et al. (2018) while incorporating the tighter confidence bounds afforded by QMC to achieve an improved quantum regret bound.

Let the vector $\boldsymbol{z}(s)$ denote the i_s-th coordinate vector. Then, the expressions $\boldsymbol{r}^U(s)\boldsymbol{z}(s)$ and $\boldsymbol{C}^L(s)\boldsymbol{z}(s)$ represent the UCB of the reward and the LCB of the resource consumption vector, respectively, for the chosen arm $i(s)$. Let $\boldsymbol{y}(s) = \boldsymbol{v}(s)/\|\boldsymbol{v}(s)\|_1$. The following equation satisfies in each stage s, where $\boldsymbol{\xi}^*$ is the optimal solution of $\mathrm{OPT}_{\mathrm{LP}}$ defined in (1a):

$$\boldsymbol{z}(s) \in \arg\max_{\|\boldsymbol{z}\|_1 = 1} \frac{\boldsymbol{r}^U(s)^\top \boldsymbol{z}}{\boldsymbol{y}(s)^\top \boldsymbol{C}^L(s) \boldsymbol{z}}, \tag{20}$$

$$\frac{\boldsymbol{r}^U(s)^\top \boldsymbol{z}(s)}{\boldsymbol{y}(s)^\top \boldsymbol{C}^L(s) \boldsymbol{z}(s)} \geq \frac{\boldsymbol{r}^U(s)^\top \boldsymbol{\xi}^*}{\boldsymbol{y}(s)^\top \boldsymbol{C}^L(s) \boldsymbol{\xi}^*}. \tag{21}$$

Summing over s, we have

$$\sum_{s=1}^{\tau-1}(\boldsymbol{r}^U(s)^\top \boldsymbol{z}(s))(\boldsymbol{y}(s)^\top \boldsymbol{C}^L(s)\boldsymbol{\xi}^*) \geq \sum_{s=1}^{\tau-1}(\boldsymbol{r}^U(s)^\top \boldsymbol{\xi}^*)(\boldsymbol{y}(s)^\top \boldsymbol{C}^L(s)\boldsymbol{z}(s)). \tag{22}$$

Then, the right side of the above equation satisfies:

$$\sum_{s=1}^{\tau-1}(\boldsymbol{r}^U(s)^\top \boldsymbol{\xi}^*)(\boldsymbol{y}(s)^\top \boldsymbol{C}^L(s)\boldsymbol{z}(s)) \geq \sum_{s=1}^{\tau-1}(\boldsymbol{r}^\top \boldsymbol{\xi}^*)(\boldsymbol{y}(s)^\top \boldsymbol{C}^L(s)\boldsymbol{z}(s)) = \mathrm{OPT}_{\mathrm{LP}} \sum_{s=1}^{\tau-1}(\boldsymbol{y}(s)^\top \boldsymbol{C}^L(s)\boldsymbol{z}(s)). \tag{23}$$

Here, the first line comes from $\boldsymbol{r}$ lies in the confidence interval, and the second line is due to $\mathrm{OPT}_{\mathrm{LP}} = \boldsymbol{r}^\top \boldsymbol{\xi}^*$.

Next, we introduce the regret bound of multiplicative weight, a folklore result in online learning.

Proposition 6 (Freund & Schapire (1997)). *For any fixed ϵ, $\forall \boldsymbol{y} \in [0,1]^d$ satisfying $\|\boldsymbol{y}\|_1 = 1$ and any sequence of $\boldsymbol{C}^L(s)\boldsymbol{z}(s)$, we have*

$$\sum_{s=1}^{\tau-1}(\boldsymbol{y}(s)^\top \boldsymbol{C}^L(s)\boldsymbol{z}(s)) \geq (1-\epsilon)\sum_{s=1}^{\tau-1}(\boldsymbol{y}^\top \boldsymbol{C}^L(s)\boldsymbol{z}(s)) - \ln d/\epsilon. \tag{24}$$

For the left-hand side of Eq. (22), we define the vector $\bar{\boldsymbol{y}} = \frac{1}{\mathrm{REW}_{\mathrm{UCB}}}\sum_s(\boldsymbol{r}^U(s)^\top \boldsymbol{z}(s))\boldsymbol{y}(s)^\top$, where $\mathrm{REW}_{\mathrm{UCB}} = \sum_s(\boldsymbol{r}^U(s)^\top \boldsymbol{z}(s))$ represents the UCB of total reward in the stages s. Consequently, we have $\bar{\boldsymbol{y}} \in [0,1]^d$, $\|\bar{\boldsymbol{y}}\|_1 = 1$. Therefore, the following inequality holds:

$$\sum_s(\boldsymbol{r}^U(s)^\top \boldsymbol{z}(s))(\boldsymbol{y}(s)^\top \boldsymbol{C}^L(s)\boldsymbol{\xi}^*) \leq \sum_s(\boldsymbol{r}^U(s)^\top \boldsymbol{z}(s))(\boldsymbol{y}(s)^\top \boldsymbol{C}\boldsymbol{\xi}^*) = \mathrm{REW}_{\mathrm{UCB}}\bar{\boldsymbol{y}}^\top \boldsymbol{C}\boldsymbol{\xi}^* \leq B \cdot \mathrm{REW}_{\mathrm{UCB}}. \tag{25}$$

After successfully deriving both the left and right sides of (22), we proceed to combine them in order to establish the upper bound on regret. Before doing so, we introduce the following proposition, which bounds the difference between the LCB of the cumulative resource consumption and the actual resource consumption. The derivation of this proposition follows the same approach as Badanidiyuru et al. (2018, Lemma 5.6). Our Proposition 5 ensure the accuracy of the classical estimator for the resource consumption.

Proposition 7 (Badanidiyuru et al. (2018, Lemma 5.6)). *Let $K = \mathcal{O}(\log(dT))$ and $\boldsymbol{c}_s$ be the random resource consumption of arm $i(s)$ at time s. Then we have the following high probability events.*

$$\|\sum_{s=1}^{\tau-1}(\boldsymbol{c}_s - \boldsymbol{C}^L(s)\boldsymbol{z}(s))\|_\infty \leq \mathcal{O}\left(\sqrt{KmB} + Km\log T\right). \tag{26}$$

Now, we combine the results of Proposition 7, (23), and (25) and apply them to (22). Denote c_s as the random resource consumption in stage s, and set $\epsilon = \sqrt{\frac{\ln d}{B}}$. We have

$$\begin{aligned}
&\mathrm{REW}_{\mathrm{UCB}} \\
&\geq \mathrm{OPT}_{\mathrm{LP}}\left(\frac{(1-\epsilon)}{B}\sum_{s=1}^{\tau-1}(\boldsymbol{y}^\top \boldsymbol{C}^L(s)\boldsymbol{z}(s)) - \frac{\ln d}{B\epsilon}\right) \\
&= \mathrm{OPT}_{\mathrm{LP}}\left(\frac{(1-\epsilon)}{B}\sum_{s=1}^{\tau-1}\boldsymbol{y}^\top \boldsymbol{c}_s - \frac{(1-\epsilon)}{B}\sum_{s=1}^{\tau-1}\boldsymbol{y}^\top(\boldsymbol{c}_s - \boldsymbol{C}^L(s)\boldsymbol{z}(s)) - \frac{\ln d}{B\epsilon}\right) \\
&\geq \mathrm{OPT}_{\mathrm{LP}}\left(\frac{(1-\epsilon)}{B}(B - 2mC_1\log T) - \frac{1}{B}\|\sum_{s=1}^{\tau-1}(\boldsymbol{c}_s - \boldsymbol{C}^L(s)\boldsymbol{z}(s))\|_\infty - \frac{\ln d}{B\epsilon}\right) \\
&\geq \mathrm{OPT}_{\mathrm{LP}}\left(\frac{(1-\epsilon)}{B}(B - 2mC_1\log T) - \frac{\ln d}{B\epsilon}\right) - \mathcal{O}\left(\sqrt{\log(dT)}\right)\left(\mathrm{OPT}_{\mathrm{LP}}\sqrt{\frac{m}{B}}\right) - \mathcal{O}\left(m\log(dT)\log T\right) \\
&\geq \mathrm{OPT}_{\mathrm{LP}}\left(1 - 2\sqrt{\frac{\ln d}{B}} - \frac{2mC_1\log T}{B}\right) - \mathcal{O}\left(\sqrt{\log(dT)}\right)\left(\mathrm{OPT}_{\mathrm{LP}}\sqrt{\frac{m}{B}}\right) - \mathcal{O}\left(m\log(dT)\log T\right),
\end{aligned} \tag{27}$$

The third inequality follows from the fact that, upon termination of the algorithm, at least one resource must be exhausted. Let us choose the fixed strategy $\boldsymbol{y}$ to be the unit vector $\boldsymbol{e}_i$ corresponding to the exhausted resource i. Then, immediately after the preparation stage, the total consumption of that resource is guaranteed to be at least $B - 2mC_1 \log T$.

Let REW denote the expected total reward, and let N_i represent the number of times the QMC method is invoked for arm i. Based on Proposition 5, the following inequality holds:

$$\begin{aligned}
\mathrm{REW} &\geq \sum_{s=1}^{\tau-1} r_{i(s)} \\
&= \sum_{s=1}^{\tau-1} (r_{i(s)} - (\boldsymbol{r}^U(s)^\top \boldsymbol{z}(s)) + (\boldsymbol{r}^U(s)^\top \boldsymbol{z}(s))) \\
&= \mathrm{REW}_{\mathrm{UCB}} - \sum_{s=1}^{\tau-1} ((\boldsymbol{r}^U(s)^\top \boldsymbol{z}(s)) - r_{i(s)}) \\
&\geq \mathrm{REW}_{\mathrm{UCB}} - \sum_{i=1}^{m} \sum_{n=1}^{N_i} \mathrm{rad}_i \frac{2C_1}{\mathrm{rad}_i/2} \log T \\
&= \mathrm{REW}_{\mathrm{UCB}} - 4C_1 \log T \sum_{i=1}^{m} N_i \\
&\geq \mathrm{REW}_{\mathrm{UCB}} - 4mC_1 \log T \log T.
\end{aligned} \tag{28}$$

Combining (27), (28), the total regret satisfies

$$\begin{aligned}
\mathrm{Regret} &\leq \mathrm{OPT}_{\mathrm{LP}} - \mathrm{REW} \\
&\leq \mathrm{OPT}_{\mathrm{LP}} \left(2\sqrt{\frac{\ln d}{B}} + \frac{2mC_1 \log T}{B} \right) + 4mC_1 \log^2 T \\
&\quad + \mathcal{O}\left(\sqrt{\log(dT)}\right) \left(\mathrm{OPT}_{\mathrm{LP}} \sqrt{\frac{m}{B}} \right) + \mathcal{O}\left(m \log(dT) \log T\right) \\
&= \mathcal{O}\left(\sqrt{\log(dT)}\right) \left(\mathrm{OPT}_{\mathrm{LP}} \sqrt{\frac{m}{B}} \right) + \mathcal{O}\left(m \log(dT) \log T\right),
\end{aligned} \tag{29}$$

where we apply the fact $m \leq B/\log(dT)$ in Eq. (29). This concludes Theorem 1.

C Proofs for Problem-Dependent Quantum BwK

In this section, we provide proof details for the problem-dependent quantum BwK algorithm, including its time complexity and regret analyses.

The parameter choices in Algorithm 2 do not require the exact values of the problem-dependent parameters δ, χ, and σ; suitable lower bounds suffice. In Phase I, the algorithm identifies the optimal arms and non-binding constraints by comparing the confidence-bound LP values $\mathrm{OPT}_{\mathrm{LP}}^L(t)$, $\mathrm{OPT}_i^U(t)$, and $\mathrm{OPT}_j^U(t)$. These identification tests use the UCB and LCB estimates of rewards and resource consumption, rather than the unknown parameters or the true optimal value $\mathrm{OPT}_{\mathrm{LP}}$.

After Phase I, the confidence estimates can be used to obtain lower bounds for parameter selection in Phase II. Specifically, on the high-probability event of correct identification in Proposition 8, we have $\hat{\mathcal{I}}^* = \mathcal{I}^*$ and $\hat{\mathcal{J}}' = \mathcal{J}'$, and hence $\mathcal{J}^* = [d] \setminus \hat{\mathcal{J}}'$. With these sets identified and the confidence bounds valid, we derive lower bounds for δ, χ, and σ as follows.

First, recall that $\mathrm{OPT}_{\mathrm{LP}}^L(t) \leq \mathrm{OPT}_{\mathrm{LP}}$, $\mathrm{OPT}_i^U(t) \geq \mathrm{OPT}_i$, and $\mathrm{OPT}_j^U(t) \geq \mathrm{OPT}_j$. By the definition of δ in (6), a lower bound is given by

$$\underline{\delta} := \frac{1}{T} \left(\mathrm{OPT}_{\mathrm{LP}}^L(t) - \max \left\{ \max_{i \in \mathcal{I}^*} \mathrm{OPT}_i^U(t), \max_{j \in \mathcal{J}'} \mathrm{OPT}_j^U(t) \right\} \right). \tag{30}$$

Second, χ does not need to be estimated separately. For any $i \in \mathcal{I}^*$, setting the i-th coordinate of $\boldsymbol{\xi}^*$ to zero gives a feasible solution to the LP defining OPT_i. Therefore,

$$\delta T \le \mathrm{OPT}_{\mathrm{LP}} - \mathrm{OPT}_i \le r_i \xi_i^* \le \xi_i^*, \tag{31}$$

where the last inequality follows from $r_i \in [0,1]$. Taking the minimum over $i \in \mathcal{I}^*$ gives $\delta \le \frac{1}{T}\min_{i\in\mathcal{I}^*} \xi_i^* = \chi$. Hence, $\underline{\delta}$ is also a lower bound for χ.

Finally, the confidence bounds satisfy $\boldsymbol{C}^L_{\mathcal{J}^*,\mathcal{I}^*}(t) \le \boldsymbol{C}_{\mathcal{J}^*,\mathcal{I}^*} \le \boldsymbol{C}^U_{\mathcal{J}^*,\mathcal{I}^*}(t)$ element-wise. By the singular-value perturbation inequality, we obtain a lower bound for σ:

$$\underline{\sigma} := \sigma_{\min}\left(\frac{\boldsymbol{C}^U_{\mathcal{J}^*,\mathcal{I}^*}(t) + \boldsymbol{C}^L_{\mathcal{J}^*,\mathcal{I}^*}(t)}{2}\right) - \frac{1}{2}\|\boldsymbol{C}^U_{\mathcal{J}^*,\mathcal{I}^*}(t) - \boldsymbol{C}^L_{\mathcal{J}^*,\mathcal{I}^*}(t)\|_2. \tag{32}$$

C.1 Proof of Proposition 2

Recall the original LP defined in Eq. (10a):

$$\mathrm{OPT}^L_{\mathrm{LP}}(t) := \max_{\boldsymbol{\xi}} \boldsymbol{r}^L(t)^\top \boldsymbol{\xi},\ \text{s.t.}\ \boldsymbol{C}^U(t)\boldsymbol{\xi} \le \boldsymbol{B},\ \boldsymbol{\xi} \ge 0,$$

which needs to be solved to $\epsilon_{\mathrm{LP}} \cdot T$-optimality and $\epsilon_{\mathrm{LP}} \cdot T$-approximation. Since existing quantum approximate LP solvers require all coefficients to lie within $[-1,1]$, we rescale the above LP by a factor of T and solve the following equivalent formulation with accuracy ϵ_{LP}:

$$\max_{\boldsymbol{\xi}'} \boldsymbol{r}^L(t)^\top \boldsymbol{\xi}',\ \text{s.t.}\ \boldsymbol{C}^U(t)\boldsymbol{\xi}' \le \boldsymbol{B}/T,\ \boldsymbol{\xi}' \ge 0. \tag{33}$$

Under this scaling, the LP coefficients satisfy the boundedness condition in $[-1,1]$, and the solution satisfies $\sum_i \xi'_i \le 1$. Let $\boldsymbol{\eta}'$ denote the dual variable; then it holds that $\sum_i \eta'_i \le 1/B$. By existing reduction from general LPs to zero-sum games (Apeldoorn & Gilyén, 2019a, Section 4), solving the LP within ϵ_{LP} accuracy requires solving the corresponding zero-sum game to additive error $\epsilon'' = (\epsilon_{\mathrm{LP}})/(6(\frac{1}{B}+1))$. The associated zero-sum game formulation is:

$$\min_{\boldsymbol{y}''\in\Delta^{n+2},\lambda\in\mathbb{R}} \lambda, \quad \text{s.t.}\ \boldsymbol{A}''\boldsymbol{y}'' \le \lambda \boldsymbol{e}, \tag{34}$$

where the matrix $\boldsymbol{A}''$ is given by:

$$\boldsymbol{A}'' = \begin{pmatrix} \boldsymbol{e}^\top & 1 & -1 \\ -\boldsymbol{e}^\top & 1 & 1 \\ -\boldsymbol{r}^L(t)^\top & 0 & \alpha \\ \boldsymbol{C}^U(t) & 0 & -\boldsymbol{B}/T \end{pmatrix}. \tag{35}$$

The quantum algorithm for solving zero-sum games achieves a time complexity of $\tilde{\mathcal{O}}\left(\sqrt{m+n}\cdot\epsilon^{-2.5}\right)$ (Bouland et al., 2023; Gao et al., 2024). Applying this algorithm to the scaled LP yields a total running time of:

$$\log(1/(\epsilon_{\mathrm{LP}}/T))\tilde{\mathcal{O}}\left(\sqrt{m+n}\frac{(1\cdot(\frac{1}{B}+1))^{2.5}}{(\epsilon_{\mathrm{LP}})^{2.5}}\right) = \tilde{\mathcal{O}}\left(\frac{\sqrt{m+n}}{\epsilon_{\mathrm{LP}}^{2.5}}\right). \tag{36}$$

The same rescaling technique with factor T can be applied to the LPs in (10b) and (10c), and with factor $T-t$ to those in (8) and (9), yielding the desired quantum time complexity in each case.

C.2 Proof of Theorem 2

The quantum time complexity in each round of Phase I is determined by two main components: the cost of QMC and the cost of quantum LP solver. For QMC, if an arm is queried N times, the quantum time complexity for estimating its expected outcome is $\tilde{\mathcal{O}}\left(N + d\log d\right)$. Since each arm is pulled $\tilde{\mathcal{O}}\left(\sqrt{d}\log T\right)$ times with high probability according to Proposition 8, and the number of pulls per arm increases geometrically across rounds, the total sampling complexity is bounded by $\sum_{i=0}^{\mathcal{O}(\log(\sqrt{d}\log T))} m(2^i + d\log d) = \mathcal{O}\left(m\sqrt{d}\log T + md\log d\log(\sqrt{d}\log T)\right)$. The quantum LP

solver in Lemma 4 requires oracle access to the coefficients of the LP instance. In each round, the algorithm solves $\mathcal{O}(m+d)$ LPs. To support quantum access to these LPs, the algorithm constructs the required oracles based on the outputs of the QMC procedure, which incurs an additional cost of $\tilde{\mathcal{O}}(md)$ for preparing the relevant QRAM structures. Combining these components, the total quantum time complexity of Phase I is:

$$\mathcal{O}(\log(\sqrt{d}\log T))\left(\tilde{\mathcal{O}}(md)+\mathcal{O}(m+d)\tilde{\mathcal{O}}\left(\frac{\sqrt{m+d}}{\epsilon_{\mathrm{LP}}^{2.5}}\right)\right)+\mathcal{O}\left(m\sqrt{d}\log T+md\log d\log(\sqrt{d}\log T)\right). \tag{37}$$

In Phase II, the algorithm solves an LP problem $\mathcal{O}(T)$ times to exhaust the binding resources. In each round, the algorithm call an quantum oracle and immediately sample it, acting as a classical sample. The estimation will be updated by the classical Monte Carlo methods in Lemma 1 in time $\mathcal{O}(d)$, and updating the QRAM also incurs a cost of $\tilde{\mathcal{O}}(d)$. Consequently, the time complexity in this phase is:

$$\mathcal{O}(T)\left(\tilde{\mathcal{O}}(d)+\tilde{\mathcal{O}}\left(\frac{\sqrt{m+d}}{\epsilon_{\mathrm{LP}}^{2.5}}\right)\right). \tag{38}$$

Combining (37) and (38), the overall time complexity of Algorithm 2 with quantum LP solver in Lemma 4 is

$$\tilde{\mathcal{O}}\left(md+\frac{(m+d)^{1.5}}{\epsilon_{\mathrm{LP}}^{2.5}}+\left(\frac{\sqrt{m+d}}{\epsilon_{\mathrm{LP}}^{2.5}}+d\right)T\right). \tag{39}$$

Time complexity of the classical algorithm in Li et al. (2021) We next analyze the time complexity of the classical two-phase algorithm proposed in Li et al. (2021), which requires solving LPs to high precision. Unlike the approximate LP solvers considered in our work, this algorithm relies on near-exact solutions. To estimate the cost, we adopt the interior-point method from Jiang et al. (2020), which solves LPs in time $\mathcal{O}(n^{2+\min\{\frac{1}{18},\omega-2,\frac{1-\alpha}{2}\}})$, where ω and α denote the matrix multiplication exponent and its dual, respectively. Under the current state-of-the-art result showing that $\omega<2.371339$ (Alman et al., 2025) and $\alpha>0.31389$ (Le Gall & Urrutia, 2018), this becomes $\mathcal{O}(n^{2.372})$.

In Phase I, the algorithm identifies the optimal arm and the corresponding binding constraints. Each round involves m arm pulls and $\mathcal{O}(m+d)$ LPs. The UCB/LCB updates require $\mathcal{O}(md)$ time per round. Since this phase runs for $\mathcal{O}(\log T)$ rounds, the total complexity is:

$$\mathcal{O}(\log T)\left(\mathcal{O}(md)+\mathcal{O}(m+d)\mathcal{O}\left(\max\{m,d\}^{2.372}\right)\right). \tag{40}$$

In Phase II, each of the $\mathcal{O}(T)$ rounds involves pulling an arm and solving a single LP.

$$\mathcal{O}(T)\left(\mathcal{O}(d)+\mathcal{O}\left(\max\{m,d\}^{2.372}\right)\right). \tag{41}$$

Combining both phases, the total time complexity of the classical algorithm in Li et al. (2021) is:

$$\mathcal{O}\left(\max\{m,d\}^{2.372}(T+m+d)\right). \tag{42}$$

C.3 Proof of Proposition 3

To prove Proposition 3, we first establish in Proposition 8 an upper bound on the number of pulls required during Phase I.

Proposition 8. *In Phase I of Algorithm 2, suppose we obtain an $\epsilon_{\mathrm{LP}}T$-optimal and $\epsilon_{\mathrm{LP}}T$-approximate solution for $\mathrm{OPT}_{\mathrm{LP}}^{\mathrm{L}}(t)$, $\mathrm{OPT}_i^{\mathrm{U}}(t)$, and $\mathrm{OPT}_j^{\mathrm{U}}(t)$, denoted as $\widetilde{\mathrm{OPT}}_{\mathrm{LP}}^{\mathrm{L}}(t)$, $\widetilde{\mathrm{OPT}}_i^{\mathrm{U}}(t)$, and $\widetilde{\mathrm{OPT}}_j^{\mathrm{U}}(t)$. To ensure that the following inequalities hold with high probability $1-\frac{2md}{T^2}$, each arm will be played no more than*

$$\tilde{\mathcal{O}}\left(\left(2+\frac{1}{b}\right)\frac{\sqrt{d}\log T}{\delta-2\epsilon_{\mathrm{LP}}}\right) \tag{43}$$

times:

$$\mathrm{OPT}_i^U(t) - \mathrm{OPT}_i < \frac{\delta T}{2} - \epsilon_{\mathrm{LP}} T \quad \text{for } i \in \mathcal{I}^*, \tag{44}$$

$$\mathrm{OPT}_j^U(t) - \mathrm{OPT}_j \leq \frac{\delta T}{2} - \epsilon_{\mathrm{LP}} T \quad \text{for } j \in \mathcal{J}', \tag{45}$$

$$\mathrm{OPT}_{\mathrm{LP}}^L(t) - \mathrm{OPT}_{\mathrm{LP}} > -\frac{\delta T}{2} + \epsilon_{\mathrm{LP}} T. \tag{46}$$

Under these conditions, the approximate solutions satisfy

$$\begin{aligned}
\widetilde{\mathrm{OPT}}_{\mathrm{LP}}^L(t) - \widetilde{\mathrm{OPT}}_i^U(t) &\geq \mathrm{OPT}_{\mathrm{LP}}^L(t) - \mathrm{OPT}_i^U(t) - 2\epsilon_{\mathrm{LP}} T > \mathrm{OPT}_{\mathrm{LP}} - \mathrm{OPT}_i - \delta T \geq 0, \\
\widetilde{\mathrm{OPT}}_{\mathrm{LP}}^L(t) - \widetilde{\mathrm{OPT}}_j^U(t) &\geq \mathrm{OPT}_{\mathrm{LP}}^L(t) - \mathrm{OPT}_j^U(t) - 2\epsilon_{\mathrm{LP}} T > \mathrm{OPT}_{\mathrm{LP}} - \mathrm{OPT}_j - \delta T \geq 0.
\end{aligned} \tag{47}$$

This implies that if the LP is approximately solved, the optimal arms $\mathcal{I}^*$ and the non-binding constraints $\mathcal{J}'$ can be correctly identified with high probability.

Proof. In each pull of arm, the reward is estimated using the univariate quantum Monte Carlo method in Lemma 2, while the estimation of the d-dimension resource consumption vector of each arm uses the multivariate quantum Monte Carlo method in Lemma 3. In the following discussion, we consider the event that all expectations for the reward and resource consumption are bounded by the corresponding UCBs and LCBs produced by the quantum Monte Carlo method. In Phase I, suppose each arm is pulled $N(t) \equiv C_2 n(t)\sqrt{\log n(t)}$ times. As a result, both the univariate and multivariate QMC methods are invoked $2m \log N(t)$ times in total. Given that the failure probability of a single QMC invocation is δ_{QMC}, this setup ensures that the true expectations fall within the estimated confidence bounds with probability at least $1 - 2m \log N(t) \cdot \delta_{\mathrm{QMC}}$.

We then show that if

$$n(t) \geq \left(2 + \frac{1}{b}\right) \frac{C_1 \sqrt{d} \log\left(d/\delta_{\mathrm{QMC}}\right)}{\delta/2 - \epsilon_{\mathrm{LP}}}, \tag{48}$$

then Eqs. (44), (45), and (46) hold.

Proof of (44). We introduce the following LP to relate $\mathrm{OPT}_i^U(t)$ with OPT_i

$$\begin{aligned}
\max_{\boldsymbol{\xi}} \quad & \boldsymbol{r}^\top \boldsymbol{\xi}, \\
\text{s.t.} \quad & \boldsymbol{C}\boldsymbol{\xi} \leq (1 + \frac{\sqrt{d} \log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}) \boldsymbol{B}, \\
& \xi_i = 0, \ \boldsymbol{\xi} \geq 0.
\end{aligned} \tag{49}$$

It is easy to see that the optimal solution of (49) is $\left(1 + \frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{bn(t)}\right) \mathrm{OPT}_i$.

Let $\boldsymbol{\xi}^U(t)$ be the optimal solution of $\mathrm{OPT}_i^U(t)$. Then we will prove that $\boldsymbol{\xi}^U(t)$ is a feasible solution of (49) and build the relation between $\mathrm{OPT}_i^U(t)$ and OPT_i.

First, we prove the feasibility

$$\begin{aligned}
(\boldsymbol{C}\boldsymbol{\xi}^U(t))_j &= (\boldsymbol{C}^L(t)\boldsymbol{\xi}^U(t))_j + ((\boldsymbol{C} - \boldsymbol{C}^L(t))\boldsymbol{\xi}^U(t))_j \\
&\leq B + \|(\boldsymbol{C} - \boldsymbol{C}^L(t))_j\|_\infty \|\boldsymbol{\xi}^U(t)\|_1 \\
&\leq B + \frac{\sqrt{d} \log\left(d/\delta_{\mathrm{QMC}}\right)}{n(t)} T, \\
&= B\left(1 + \frac{\sqrt{d} \log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}\right).
\end{aligned} \tag{50}$$

Then we have

$$\begin{aligned}
\mathrm{OPT}_i^U(t) &= \boldsymbol{r}^U(t)^\top \boldsymbol{\xi}^U(t) \\
&\le \boldsymbol{r}^\top \boldsymbol{\xi}^U(t) + \frac{C_1 \log(1/\delta_{\mathrm{QMC}})}{N(t)} T \\
&\le \left(1 + \frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{bn(t)}\right) \mathrm{OPT}_i + \frac{C_1 \log(1/\delta_{\mathrm{QMC}})}{N(t)} T \\
&< \mathrm{OPT}_i + \left(1 + \frac{1}{b}\right) \frac{C_1 \sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{n(t)} T \\
&\le \mathrm{OPT}_i + (\frac{\delta}{2} - \epsilon_{\mathrm{LP}}) T.
\end{aligned} \tag{51}$$

Hence we have established (44).

Proof of (45). Let $\boldsymbol{\eta}$ be the optimal solution to OPT_j, we will prove that

$$\boldsymbol{\eta}'(t) = \boldsymbol{\eta} + \left(\frac{C_1 \log(1/\delta_{\mathrm{QMC}})}{bN(t)} + \left(2 + \frac{1}{b}\right) \frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{bn(t)}\right) (1, 0, \ldots, 0)^\top \tag{52}$$

is the feasible solution to OPT_j^U. We have

$$\begin{aligned}
&{\boldsymbol{C}^L}^\top \boldsymbol{\eta}'(t) \\
&= \boldsymbol{C}^L(t)^\top \boldsymbol{\eta} + \frac{C_1 \log(1/\delta_{\mathrm{QMC}})}{bN(t)} \boldsymbol{C}^L(t)^\top (1, 0, \ldots, 0)^\top + \left(2 + \frac{1}{b}\right) \frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{bn(t)} \boldsymbol{C}^L(t)^\top (1, 0, \ldots, 0)^\top \\
&= \boldsymbol{C}^\top \boldsymbol{\eta} + (\boldsymbol{C}^L(t) - \boldsymbol{C})^\top \boldsymbol{\eta} + \frac{C_1 \log(1/\delta_{\mathrm{QMC}})}{N(t)} \mathbf{1} + \left(2 + \frac{1}{b}\right) \frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{n(t)} \mathbf{1} \\
&\ge \boldsymbol{r} + \boldsymbol{C}_{j,\cdot}^\top - \max_{j \in [d]} \|\boldsymbol{C}_{j,\cdot}^L(t) - \boldsymbol{C}_{j,\cdot}\|_\infty (\mathbf{1}^\top \boldsymbol{\eta}) \mathbf{1} + \frac{C_1 \log(1/\delta_{\mathrm{QMC}})}{N(t)} \mathbf{1} + \left(2 + \frac{1}{b}\right) \frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{n(t)} \mathbf{1} \\
&\ge \boldsymbol{r} + \boldsymbol{C}_{j,\cdot}^\top - \frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{n(t)} \left(1 + \frac{1}{b}\right) \mathbf{1} + \frac{C_1 \log(1/\delta_{\mathrm{QMC}})}{N(t)} \mathbf{1} + \left(2 + \frac{1}{b}\right) \frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{n(t)} \mathbf{1} \\
&= \boldsymbol{r} + \boldsymbol{C}_{j,\cdot}^\top + \frac{C_1 \log(1/\delta_{\mathrm{QMC}})}{N(t)} \mathbf{1} + \frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{n(t)} \mathbf{1} \\
&\ge \boldsymbol{r}^U(t) + (\boldsymbol{C}_{j,\cdot}^U(t))^\top.
\end{aligned} \tag{53}$$

The forth line is due to $\boldsymbol{B}^\top \boldsymbol{\eta} - B \le T$ and $\mathbf{1}^\top \boldsymbol{\eta} \le 1 + \frac{1}{b}$. As a result, the following inequalities hold:

$$\begin{aligned}
\mathrm{OPT}_j^U(t) &\le \boldsymbol{B}^\top \boldsymbol{\eta}'(t) - B \\
&\le \boldsymbol{B}^\top \boldsymbol{\eta} - B + \left(\frac{C_1 \log(1/\delta_{\mathrm{QMC}})}{N(t)} + \left(2 + \frac{1}{b}\right) \frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{n(t)}\right) T \\
&\le \boldsymbol{B}^\top \boldsymbol{\eta} - B + \left(2 + \frac{1}{b}\right) \frac{C_1 \sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{n(t)} T \\
&\le \mathrm{OPT}_j + \left(\frac{\delta}{2} - \epsilon_{\mathrm{LP}}\right) T.
\end{aligned} \tag{54}$$

Hence we have established (45).

Proof of (46). Recall that $\boldsymbol{\xi}^*$ is the optimal solution of $\mathrm{OPT}_{\mathrm{LP}}$, we will prove that $\left(1-\frac{\sqrt{d}\log(d/\delta_{\mathrm{QMC}})}{bn(t)}\right)\boldsymbol{\xi}^*$ is the feasible solution to $\mathrm{OPT}^L_{\mathrm{LP}}(t)$.

$$\begin{aligned}
&B-\left(1-\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}\right)(\boldsymbol{C}^U(t)\boldsymbol{\xi}^*)_j\\
&=B-(\boldsymbol{C}^U(t)\boldsymbol{\xi}^*)_j+\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{n(t)}(\boldsymbol{C}^U(t)\boldsymbol{\xi}^*)_j\\
&\geq B-(\boldsymbol{C}\boldsymbol{\xi}^*)_j-((\boldsymbol{C}^U(t)-\boldsymbol{C})\boldsymbol{\xi}^*)_j+\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}(\boldsymbol{C}\boldsymbol{\xi}^*)_j\\
&\geq B-(\boldsymbol{C}\boldsymbol{\xi}^*)_j-\|(\boldsymbol{C}^U(t)-\boldsymbol{C})_{j,\cdot}\|_\infty\|\boldsymbol{\xi}^*\|_1+\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}(\boldsymbol{C}\boldsymbol{\xi}^*)_j\\
&\geq B-(\boldsymbol{C}\boldsymbol{\xi}^*)_j-\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}B+\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}(\boldsymbol{C}\boldsymbol{\xi}^*)_j\\
&=\left(1-\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}\right)(B-(\boldsymbol{C}\boldsymbol{\xi}^*)_j)\\
&\geq 0.
\end{aligned}\tag{55}$$

Then we have the following inequality

$$\begin{aligned}
\mathrm{OPT}^L_{\mathrm{LP}}(t)&\geq\left(1-\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}\right)(\boldsymbol{r}^L(t))^\top\boldsymbol{\xi}^*\\
&=\boldsymbol{r}^\top\boldsymbol{\xi}^*+\left(\boldsymbol{r}^L(t)-\boldsymbol{r}\right)^\top\boldsymbol{\xi}^*-\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}\boldsymbol{r}^L(t)^\top\boldsymbol{\xi}^*\\
&\geq\mathrm{OPT}_{\mathrm{LP}}-\|\boldsymbol{r}^L(t)-\boldsymbol{r}\|_\infty\|\boldsymbol{\xi}^*\|_1-\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}\boldsymbol{r}^\top\boldsymbol{\xi}^*\\
&\geq\mathrm{OPT}_{\mathrm{LP}}-\frac{C_1\log\left(1/\delta_{\mathrm{QMC}}\right)}{N(t)}T-\frac{\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{bn(t)}T\\
&>\mathrm{OPT}_{\mathrm{LP}}-\left(1+\frac{1}{b}\right)\frac{C_1\sqrt{d}\log\left(d/\delta_{\mathrm{QMC}}\right)}{n(t)}T\\
&>\mathrm{OPT}_{\mathrm{LP}}-\left(\frac{\delta}{2}-\epsilon_{\mathrm{LP}}\right)T.
\end{aligned}\tag{56}$$

Hence we have established (46).

In conclusion, let $\delta_{\mathrm{QMC}}=\frac{d}{T^3}$ in (48). Then, with probability at least $1-\frac{2md}{T^2}$, if $n(t)\geq\left(2+\frac{1}{b}\right)\frac{3C_1\sqrt{d}\log T}{\delta/2-\epsilon_{\mathrm{LP}}}$, then Eqs. (44), (45), and (46) are all satisfied. This completes the proof of Proposition 8. □

The result guarantees that, under the assumption $\epsilon_{\mathrm{LP}}\leq\frac{\delta}{4}$, each arm is pulled at most $\tilde{\mathcal{O}}\left(\left(2+\frac{1}{b}\right)\frac{\sqrt{d}\log T}{\delta}\right)$ times during Phase I of Algorithm 2. To upper bound the regret incurred in Phase I, we introduce the following general regret bound for BwK algorithms:

Proposition 9 (Li et al. (2021, Proposition 1))**.** *The following inequality holds for any BwK algorithm:*

$$\mathrm{Regret}\leq\sum_{i\in\mathcal{I}'}n_i(t)\Delta_i+\mathbb{E}\left[\boldsymbol{B}^{(\tau)}\right]^\top\boldsymbol{\eta}^*,\tag{57}$$

where $\Delta_i=\boldsymbol{C}_{\cdot,i}\boldsymbol{\eta}^*-r_i$ *for* $i\in[m]$.

This bound indicates that the regret consists of two parts: the cumulative contribution from pulling suboptimal arms, and the expected value of leftover resources weighted by the optimal dual solution.

In our two-phase quantum algorithm, suboptimal arms are pulled only in Phase I. Phase II selects only optimal arms and thus contributes regret solely through the second term. Observe that $\Delta_i = \boldsymbol{C}_{\cdot,i}^\top \boldsymbol{\eta}^* - r_i \leq \mathbf{1}^\top \boldsymbol{\eta}^*$, and by duality, $\boldsymbol{B}^\top \boldsymbol{\eta}^* \leq T$, implying $\mathbf{1}^\top \boldsymbol{\eta}^* \leq \frac{1}{b}$. Therefore, we obtain the bound

$$\sum_{i \in \mathcal{I}'} n_i(t) \Delta_i = \tilde{\mathcal{O}}\left(\left(2 + \frac{1}{b}\right) \frac{m\sqrt{d} \log T}{b\delta} \right), \tag{58}$$

which completes the proof of Proposition 3.

C.4 Robustness Analysis of Phase II of Algorithm 2 under Approximate LP Solutions

When addressing linear programming problems whose time complexity depends on $1/\epsilon$ rather than $\log(1/\epsilon)$, it is critical to assess how approximations might affect the algorithm's robustness. In this work, we present a regret analysis for Phase II of Algorithm 2 under approximate LP solutions. Our analysis demonstrates that even when the LP is solved to $\epsilon_{\mathrm{LP}}(T-t)/\log^2 T$-optimal and $\epsilon_{\mathrm{LP}}(T-t)/\log^2 T$-feasible, the regret in Phase II remains a problem-dependent constant, provided that ϵ_{LP} is sufficiently small. Such robustness of LP may be of independent interest. Our analysis is built upon the framework of the pure classical problem-dependent bandit with knapsacks problem as presented in Li et al. (2021). In particular, with $\theta = \min\left\{ \frac{\min\{1,\sigma^2\}\min\{\chi,\delta\}}{12\min\{m^2,d^2\}}, \left(2+\frac{1}{b}\right)^{-2} \cdot \frac{\delta}{5} \right\}$ and $\epsilon = \frac{\min\{1,\sigma\}\min\{\chi,\delta\}b}{5d^{3/2}}$, Phase II of Algorithm 2 can be decomposed into three stages:

1. In initial stage, the optimal arms have not yet been played for a sufficient number of rounds, so that the condition $\|\boldsymbol{C}_{\cdot,i}^L(t) - \boldsymbol{C}_{\cdot,i}\|_\infty > \theta$ holds. Intuitively, the duration of this stage is of order $\mathcal{O}(\log T)$.
2. After initial stage, each optimal arm has been sampled sufficiently often, which facilitates an accurate estimation of the resource consumption matrix $\boldsymbol{C}$. This improved estimation ensures that the condition $\|\boldsymbol{C}_{\cdot,i}^L(t) - \boldsymbol{C}_{\cdot,i}\|_\infty \leq \theta$ continues to hold, thereby guaranteeing the stability of the smallest singular value of $\boldsymbol{C}^L(t)$. Concurrently, we impose that in each round the average remaining resource of binding constraints is maintained within the interval $[\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]$. Specifically, if $\boldsymbol{B}_{\mathcal{I}^*}^{(t-1)}/(T-t+1) \in [\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]$, then it follows with high probability that in the subsequent round t, the condition $\boldsymbol{B}_{\mathcal{I}^*}^{(t)}/(T-t) \in [\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]$ will be satisfied. Intuitively, this stage is expected to extend over $\mathcal{O}(T)$ rounds.
3. The final stage occurs when the average remaining resource deviates from the target interval, i.e., when $\frac{\boldsymbol{B}_{\mathcal{I}^*}^{(t)}}{T-t} \notin [\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]$. This stage occurs with a small probability and is expected to last only $\mathcal{O}(1)$ rounds.

Since all optimal arms have been identified in Phase I, the regret incurred during Phase II arises from the residual resources associated with the binding constraints, as characterized by Proposition 9 in Appendix C.3. In the following analysis, we examine the remaining resources of the binding constraints after stage 2, which provides an upper bound on the final residual resources after the completion of all three stages. Our analysis is conducted under the Assumption 2, which is also adopted in Li et al. (2021).

Assumption 2. *We make the following assumption on the following analysis:*

1. $\boldsymbol{r}^U(t) \geq \boldsymbol{r}$, $\boldsymbol{C}^L(t) \leq \boldsymbol{C}$ *hold element-wise for all* $t \in [T]$.
2. *all arms are optimal and all the constraints are binding, i.e.,* $\mathcal{I}^* = [m]$, $\mathcal{J}^* = [d]$.

Based on Lemma 1, the first part of Assumption 2 is satisfied with high probability. Note that in this phase, our Algorithm 2 employs the quantum oracle as a classical oracle, and uses the classical Monte Carlo method to estimate the UCB and LCB of the reward and resource consumption. Let $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{C}}_{\cdot,i}$ denote the classical estimates of the reward and consumption of arm i. If this arm is pulled N times, we have $|\hat{r}_i - r_i| \leq \sqrt{\frac{2\log T}{N}}$ and $\|\hat{\boldsymbol{C}}_{\cdot,i} - \boldsymbol{C}_{\cdot,i}\|_\infty \leq \sqrt{\frac{2\log T}{N}}$ with probability at least $1 - \frac{4md}{T^2}$ by Li et al. (2021, Lemma 2). The second part of Assumption 2 is motivated by

the intuition that the non-binding constraints will not be exhausted until the binding constraints are depleted. Under this assumption, m and d can be viewed as the numbers of optimal arms and binding constraints. The relaxation of the second part follows directly from Li et al. (2021, Appendix C4).

C.4.1 Stage 1 of Phase II

We begin by formulating the following three linear programming problems only for arms $i \in \mathcal{I}^*$.

$$\begin{aligned} \tilde{\mathrm{OPT}}_i^U(t) = \max_{\boldsymbol{\xi}} \ & \left(\boldsymbol{\mu}^U(t-1)\right)^\top \boldsymbol{\xi}, \\ \text{s.t. } & \boldsymbol{C}^L(t-1)\boldsymbol{\xi} \le \boldsymbol{B}^{(t-1)}, \\ & \xi_i = 0,\ \boldsymbol{\xi} \ge \mathbf{0}, \end{aligned} \tag{59}$$

$$\begin{aligned} \tilde{\mathrm{OPT}}_i^{U,\mathrm{Relaxed}}(t) = \max_{\boldsymbol{\xi}} \ & \left(\boldsymbol{\mu}^U(t-1)\right)^\top \boldsymbol{\xi}, \\ \text{s.t. } & \boldsymbol{C}^L(t-1)\boldsymbol{\xi} \le \boldsymbol{B}^{(t-1)} + \epsilon_1, \\ & \xi_i = 0,\ \boldsymbol{\xi} \ge \mathbf{0}, \end{aligned} \tag{60}$$

$$\begin{aligned} \tilde{\mathrm{OPT}}_{\mathrm{LP}}^U(t) = \max_{\boldsymbol{\xi}} \ & \left(\boldsymbol{\mu}^U(t-1)\right)^\top \boldsymbol{\xi}, \\ \text{s.t. } & \boldsymbol{C}^L(t-1)\boldsymbol{\xi} \le \boldsymbol{B}^{(t-1)}, \\ & \boldsymbol{\xi} \ge \mathbf{0}. \end{aligned} \tag{61}$$

Lemma 6 (Li et al. (2021, Appendix C3)). *There is a lower bound on the gap between* $\tilde{\mathrm{OPT}}_{\mathrm{LP}}^U(t)$ *and* $\tilde{\mathrm{OPT}}_i^U(t)$ *for* $t \le \frac{b\min\{\chi\theta,\delta\theta\}}{90d}T$*:*

$$\tilde{\mathrm{OPT}}_{\mathrm{LP}}^U(t) - \tilde{\mathrm{OPT}}_i^U(t) \ge \frac{\min\{\chi\theta,\delta\theta\}}{45}(T-t).$$

Now, let us define $\epsilon_1 := \epsilon_{\mathrm{LP}}(T-t)/\log^2 T$. We consider the scenario where the linear program in (61) is solved up to an ϵ_1-optimal and ϵ_1-approximate solution. Denote this solution by $\tilde{\boldsymbol{\xi}}$, which satisfies:

$$\tilde{\mathrm{OPT}}_{\mathrm{LP}}^U(t) - \epsilon_1 \le \left(\boldsymbol{\mu}^U(t-1)\right)^\top \tilde{\boldsymbol{\xi}} \le \tilde{\mathrm{OPT}}_{\mathrm{LP}}^U(t) + \epsilon_1,\ \boldsymbol{C}^L(t-1)\tilde{\boldsymbol{\xi}} \le \boldsymbol{B}^{(t-1)} + \epsilon_1,\ \tilde{\boldsymbol{\xi}} \ge \mathbf{0}. \tag{62}$$

Our goal is to derive an element-wise lower bound for the approximate solution $\tilde{\boldsymbol{\xi}}$, in order to demonstrate that even with approximate LP solutions, stage 1 can still be completed in $\mathcal{O}(\log T)$ rounds. Observe that $\tilde{\boldsymbol{\xi}}$ with $\tilde{\boldsymbol{\xi}}_i = 0$ is feasible for the relaxed version of the LP in (60). Hence, we have the inequality:

$$\left(\boldsymbol{\mu}^U(t-1)\right)^\top \tilde{\boldsymbol{\xi}} = \left(\boldsymbol{\mu}^U(t-1)\right)_{\neq i}^\top \tilde{\boldsymbol{\xi}}_{\neq i} + \left(\boldsymbol{\mu}^U(t-1)\right)_i^\top \tilde{\boldsymbol{\xi}}_i \le \tilde{\mathrm{OPT}}_i^{U,\mathrm{Relaxed}}(t) + \tilde{\boldsymbol{\xi}}_i. \tag{63}$$

Let $\boldsymbol{\xi}$ be an optimal solution to (60). Then, the scaled vector $\boldsymbol{\xi}/(1+\epsilon_1/\max\{\boldsymbol{B}^{(t-1)}\})$ is a feasible solution for (59), implying the bound:

$$\frac{\tilde{\mathrm{OPT}}_i^{U,\mathrm{Relaxed}}(t)}{1+\epsilon_1/\max\{\boldsymbol{B}^{(t-1)}\}} \le \tilde{\mathrm{OPT}}_i^U(t). \tag{64}$$

Combining the inequalities above, we obtain:

$$\begin{aligned} \tilde{\boldsymbol{\xi}}_i &\ge \tilde{\mathrm{OPT}}_{\mathrm{LP}}^U(t) - \epsilon_1 - \tilde{\mathrm{OPT}}_i^U(t)\left(1+\epsilon_1/\max\{\boldsymbol{B}^{(t-1)}\}\right) \\ &\ge \tilde{\mathrm{OPT}}_{\mathrm{LP}}^U(t) - \tilde{\mathrm{OPT}}_i^U(t) - \epsilon_1\left(1+(T-t)/\max\{\boldsymbol{B}^{(t-1)}\}\right) \\ &\ge \frac{\min\{\chi\theta,\delta\theta\}}{45}(T-t) - \frac{\epsilon_{\mathrm{LP}}(T-t)}{\log^2 T}\left(1+\frac{T-t}{(T-t)^{\frac{4}{5}}b}\right) \\ &\ge \frac{\min\{\chi\theta,\delta\theta\}}{45}(T-t) - \frac{9\epsilon_{\mathrm{LP}}(T-t)}{4b\log^2 T} \\ &\ge \frac{\min\{\chi\theta,\delta\theta\}}{90}(T-t), \end{aligned} \tag{65}$$

where the last line follows from the assumption that $\epsilon_{\text{LP}} \leq \frac{2b\log^2 T\min\{\chi\theta,\delta\theta\}}{405}$ in Proposition 4. Under this condition, the probability of each arm being played is lower bounded by $\frac{\min\{\chi\theta,\delta\theta\}}{90}$. Following a similar argument as in Appendix C.4.2, we can conclude that with high probability after $\mathcal{O}(\log T)$ rounds, the condition $\|\boldsymbol{C}\cdot, i^L(t) - \boldsymbol{C}\cdot, i\|_\infty \leq \theta$ is satisfied.

C.4.2 Stage 2 of Phase II

In Stage 2 of Phase II, the condition $\|\boldsymbol{C}^L_{\cdot,i}(t) - \boldsymbol{C}_{\cdot,i}\|_\infty \leq \theta$ is met. According to Proposition 9, the regret of the Algorithm 2 is determined by the term $\mathbb{E}\left[\boldsymbol{B}^{(\tau)}\right]^\top \boldsymbol{\eta}^*$, where $\boldsymbol{B}^{(t)}$ represents the vector of remaining resources at time t, and τ is the algorithm's stopping time. To facilitate the analysis, we define the average remaining resources as $\boldsymbol{b}^{(t)} := \frac{\boldsymbol{B}^{(t)}}{T-t}$. We then define τ' as the first time step at which $\boldsymbol{b}^{(t)}$ exits the interval $[\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]$, which marks the end of Stage 2. The value of parameter ϵ is specified in Lemma 7.

$$\tau' := \min\{t : \boldsymbol{b}^{(t)} \notin [\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]\}. \tag{66}$$

It is straightforward that with ϵ small enough, $\tau' < \tau$. In the following analysis, we focus on bounding the expectation $\mathbb{E}[T-\tau']$ to bind the upper bound of $\mathbb{E}\left[\boldsymbol{B}^{(\tau)}\right]$.

For each $t \in [T]$, we define the following event. Let $\mathcal{H}_{t-1} = (\boldsymbol{r}_s, \boldsymbol{C}_s)_{s=1}^{t-1}$ denote the filtration up to time t, which captures all reward and resource consumption observations up to round $t-1$. Define the event $\mathcal{E}_t$ as follows: it occurs if for all $s \in [t-1]$, the average remaining resource $\boldsymbol{b}^{(s)}$ stays within the interval $[\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]$, and at time t, the expected resource consumption $\mathbb{E}[\boldsymbol{C}_{\cdot,i_t}(\boldsymbol{b}^{(t-1)})|\mathcal{H}_{t-1}]$ deviates from $\boldsymbol{b}^{(t-1)}$ by more than ϵ_t in the ℓ_∞ norm. The parameter ϵ_t will be formally specified in Lemma 7.

$$\mathcal{E}_t := \left\{\boldsymbol{b}^{(s)} \in [\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon] \text{ for each } s \in [t-1] \text{ and } \|\mathbb{E}[\boldsymbol{C}_{\cdot,i_t}(\boldsymbol{b}^{(t-1)})|\mathcal{H}_{t-1}] - \boldsymbol{b}^{(t-1)}\|_\infty > \epsilon_t\right\}. \tag{67}$$

To analyze $\mathbb{E}[T-\tau']$, we examine the probability $\mathbb{P}(\tau' \leq t)$, which can be decomposed into two parts. Here, the complement of the event $\mathcal{E}_t$, denoted $\mathcal{E}_t^\complement$, is defined over the space of t vectors in $\mathbb{R}^d$.

$$\mathbb{P}(\tau' \leq t) = \mathbb{P}\left(\tau' \leq t, \bigcap_{s=1}^{T} \mathcal{E}_t^\complement\right) + \mathbb{P}\left(\tau' \leq t, \bigcup_{s=1}^{T} \mathcal{E}_t\right) \leq \mathbb{P}\left(\tau' \leq t, \bigcap_{s=1}^{T} \mathcal{E}_t^\complement\right) + \mathbb{P}\left(\bigcup_{s=1}^{T} \mathcal{E}_t\right). \tag{68}$$

Our analysis of the impact of the approximate LP solutions on these two probability terms relies on Lemma 7 and Lemma 8. We present these lemmas below and defer their proofs to the end of the subsection.

Lemma 7. *Let* $\alpha = \frac{\min\{1,\sigma\}\min\{\chi,\delta\}b}{d^{3/2}}$, $\epsilon = \frac{\alpha}{5}$,

$$\epsilon_t = \begin{cases} \frac{6}{5} & t < \frac{\alpha T}{19} \\ \frac{19b\min\{m,d\}\sqrt{d}}{\sigma}\sqrt{\frac{\log T}{t}} + \frac{4\min\{m^{\frac{5}{2}}, d^{\frac{5}{2}}\}\epsilon_{\text{LP}}}{\sigma\log^2 T} & t \geq \frac{\alpha T}{19} \end{cases}. \tag{69}$$

If $\epsilon_{\text{LP}} \leq \frac{\sigma\chi\log^2 T}{40\min\{m^{\frac{3}{2}}, d^{\frac{3}{2}}\}}$, *then there exists a constant* $\underline{T}$ *such that for all* $T > \underline{T}$

$$\mathbb{P}\left(\bigcup_{s=1}^{T} \mathcal{E}_t\right) \leq \frac{\min\{m,d\}}{T^3}. \tag{70}$$

Lemma 8. *If* $\epsilon, \bar{\epsilon}, \alpha$ *and* ϵ_t *satisfy that* $\epsilon_t = \bar{\epsilon}$ *for all* $t \leq \alpha T$ *and*

$$\frac{\alpha\bar{\epsilon}}{1-\alpha} + \sum_{t=\alpha T+1}^{T-1} \frac{\epsilon_t}{T-t} \leq \frac{2\epsilon}{3}, \tag{71}$$

then the probability is bounded by

$$\mathbb{P}\left(\tau' \leq t, \bigcap_{s=1}^{T} \mathcal{E}_s^\complement\right) \leq 2d\exp\left(-\frac{(T-t-1)\epsilon^2}{18}\right). \tag{72}$$

Based on Lemma 7 and Lemma 8, we have

$$\begin{aligned}\mathbb{P}\left(\tau' \le t\right) &\le \mathbb{P}\left(\tau' \le t, \bigcap_{s=1}^{t} \mathcal{E}_t^{\mathsf{C}}\right) + \mathbb{P}\left(\bigcup_{s=1}^{t} \mathcal{E}_t\right) \\ &\le 2d \exp\left(-\frac{(T-t-1)\min\{1,\sigma^2\}\min\{\chi^2,\delta^2\}b^2}{450d^3}\right) + \frac{\min\{m,d\}}{T^3}.\end{aligned} \tag{73}$$

Consequently, the expectation can be bounded as

$$\begin{aligned}\mathbb{E}[T-\tau'] &= \sum_{t=1}^{T} \mathbb{P}(\tau' \le t) \\ &\le \sum_{t=1}^{T} 2d \exp\left(-\frac{(T-t-1)\min\{1,\sigma^2\}\min\{\chi^2,\delta^2\}b^2}{450d^3}\right) + \frac{\min\{m,d\}}{T^3} \\ &\le \frac{900d^4}{\min\{1,\sigma^2\}\min\{\chi^2,\delta^2\}b^2} + \frac{\min\{m,d\}}{T^2}.\end{aligned} \tag{74}$$

As a result,

$$\mathbb{E}[\boldsymbol{B}^{(\tau)}] \le \mathbb{E}[\boldsymbol{B}^{(\tau')}] \le \mathbb{E}\left[(T-\tau')(\boldsymbol{b}+\epsilon\mathbf{1})\right] \le \mathbb{E}\left[(T-\tau')\frac{6}{5}\boldsymbol{b}\right] = \mathcal{O}\left(\frac{d^4}{\min\{1,\sigma^2\}\min\{\chi^2,\delta^2\}b}\right). \tag{75}$$

Using the fact that $\mathbf{1}^\top \boldsymbol{\eta}^* \le \frac{1}{b}$, we derive the expected regret bound $\mathbb{E}[\boldsymbol{B}^{(\tau)}]\boldsymbol{\eta}^* \le \mathcal{O}\left(\frac{d^4}{\min\{1,\sigma^2\}\min\{\chi^2,\delta^2\}b^2}\right)$ for Phase II of Algorithm 2, thereby completing the proof of Proposition 4.

Proof of Lemma 7 The key to the proof of Lemma 7 is to establish a lower bound on the number of times each optimal arm is played. To this end, we first invoke Lemma 9, which provides a lower bound on the probability of selecting each optimal arm when the LP problems are solved nearly exactly.

Lemma 9 (Li et al. (2021, Lemma 7))**.** *With $\epsilon \le \frac{\min\{\chi,\delta\}}{5d^{3/2}}$ and Assumption 2, if $\boldsymbol{b}^{(t)} \in [\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]$, the optimal solution of LP (8) at time $t+1$ is*

$$\boldsymbol{\eta}^*(t) = (\boldsymbol{C}^L(t))^{-1}\boldsymbol{B}^{(t)}, \tag{76}$$

and the normalized probability $\widetilde{\boldsymbol{\eta}}(t)$ used to play arm at time $t+1$ satisfies

$$\widetilde{\boldsymbol{\eta}}(t) = (\boldsymbol{C}^L(t))^{-1}\boldsymbol{b}^{(t)}, \tag{77}$$

and is element-wise lower bounded by

$$\widetilde{\boldsymbol{\eta}}(t) \ge \frac{\chi}{5}. \tag{78}$$

In our algorithm, the LP problems are solved approximately. Lemma 10 demonstrates that if each LP is solved to within an additive error of at most $\epsilon_{\mathrm{LP}}(T-t)/\log^2 T$, and $\epsilon_{\mathrm{LP}} \le \frac{\sigma\chi\log^2 T}{40\min\{m^{3/2},d^{3/2}\}}$, then the resulting approximate solution remains close to the exact one in terms of the normalized probability.

Lemma 10. *If the LP problem is solved $\epsilon_{\mathrm{LP}}(T-t)/\log^2 T$ optimal and $\epsilon_{\mathrm{LP}}(T-t)/\log^2 T$-approximate, let the approximate solution to the LP be denoted by $\boldsymbol{\eta}^{*\prime}(t)$, and define the probability of playing each arm as $\hat{\boldsymbol{\eta}}(t) = \boldsymbol{\eta}^{*\prime}(t)/|\boldsymbol{\eta}^{*\prime}(t)|_1$. Then, if $\epsilon_{\mathrm{LP}} \le \frac{\sigma\chi\log^2 T}{40\min\{m^{\frac{3}{2}},d^{\frac{3}{2}}\}}$, the element-wise difference in the probabilities of playing each arm between the approximate and exact solutions to the LP is bounded by*

$$\|\tilde{\boldsymbol{\eta}}(t) - \hat{\boldsymbol{\eta}}(t)\|_\infty \le \frac{4\min\{m^{\frac{3}{2}},d^{\frac{3}{2}}\}\epsilon_{\mathrm{LP}}}{\sigma\log^2 T} \le \frac{\chi}{10}. \tag{79}$$

Proof. By obtaining an approximate solution to the LP,

$$\|\boldsymbol{C}^L(t)(\boldsymbol{\eta}^{*\prime}(t)-\boldsymbol{\eta}^*(t))\|_\infty \le \epsilon_{\mathrm{LP}}(T-t)/\log^2 T \tag{80}$$

then

$$\begin{aligned}\|\boldsymbol{\eta}^{*\prime}(t)-\boldsymbol{\eta}^*(t)\|_\infty &\le \|\boldsymbol{\eta}^{*\prime}(t)-\boldsymbol{\eta}^*(t)\|_2\\ &\le \frac{1}{\sigma_{\min}(\boldsymbol{C}^L(t))}\|\boldsymbol{C}^L(t)(\boldsymbol{\eta}^{*\prime}(t)-\boldsymbol{\eta}^*(t))\|_2\\ &\le \frac{\sqrt{\min\{m,d\}}}{\sigma_{\min}(\boldsymbol{C}^L(t))}\|\boldsymbol{C}^L(t)(\boldsymbol{\eta}^{*\prime}(t)-\boldsymbol{\eta}^*(t))\|_\infty\\ &\le \frac{2\sqrt{\min\{m,d\}}\epsilon_{\mathrm{LP}}(T-t)}{\sigma\log^2 T},\end{aligned} \tag{81}$$

and the 1-norm of $\boldsymbol{\eta}^{*\prime}(t)$ satisfies $|\|\boldsymbol{\eta}^{*\prime}(t)\|_1-\|\boldsymbol{\eta}^*(t)\|_1| \le \min\{m,d\}\|\boldsymbol{\eta}^{*\prime}(t)-\boldsymbol{\eta}^*(t)\|_\infty$. Therefore we can bound the difference in the probabilities of playing each arm:

$$\begin{aligned}\|\tilde{\boldsymbol{\eta}}(t)-\hat{\boldsymbol{\eta}}(t)\|_\infty &= \|\frac{\boldsymbol{\eta}^{*\prime}(t)}{\|\boldsymbol{\eta}^{*\prime}(t)\|_1}-\frac{\boldsymbol{\eta}^*(t)}{\|\boldsymbol{\eta}^*(t)\|_1}\|_\infty\\ &\le \frac{(T-t)\|\boldsymbol{\eta}^{*\prime}(t)-\boldsymbol{\eta}^*(t)\|_\infty}{(T-t)\|\boldsymbol{\eta}^{*\prime}(t)\|_1}+\frac{\|\boldsymbol{\eta}^*(t)\|_\infty\left(T-t-\|\boldsymbol{\eta}^{*\prime}(t)\|_1\right)}{(T-t)\|\boldsymbol{\eta}^{*\prime}(t)\|_1}\\ &\le \frac{(1+\min\{m,d\})\frac{2\min\{m^{\frac12},d^{\frac12}\}\epsilon_{\mathrm{LP}}}{\sigma\log^2 T}}{1-\frac{2\min\{m^{\frac32},d^{\frac32}\}\epsilon_{\mathrm{LP}}}{\sigma\log^2 T}}\\ &\le \frac{4\min\{m^{\frac32},d^{\frac32}\}\epsilon_{\mathrm{LP}}}{\sigma\log^2 T},\end{aligned} \tag{82}$$

where we apply $\frac{2\min\{m^{\frac32},d^{\frac32}\}\epsilon_{\mathrm{LP}}}{\sigma\log^2 T} \le \frac{\chi}{20} < \frac12$ in the last line. □

From Lemma 9, we know that if the LP problem is solved approximately, the approximate probability is element-wise lower bounded by $\hat{\boldsymbol{\eta}}(t) \ge \tilde{\boldsymbol{\eta}}(t) - \|\tilde{\boldsymbol{\eta}}(t)-\hat{\boldsymbol{\eta}}(t)\|_\infty \ge \frac{\chi}{10}$. This implies that each optimal arm is selected with probability at least $\frac{\chi}{10}$. In the following, we show that if $\boldsymbol{b}^{(s)} \in [\boldsymbol{b}-\epsilon, \boldsymbol{b}+\epsilon]$ for all $s \in [t]$ and $t \ge \frac{3200\log T}{\chi^2}$, then the number of times arm $i \in \mathcal{I}^*$ is played up to time t, denoted as $n_i(t)$, is lower bounded by $n_i(t) \ge \frac{\chi t}{20}$.

Recall that the filtration $\mathcal{H}_t$ contains all random rewards and resource consumption observations up to time t, i.e., $\mathcal{H}_t = \{\boldsymbol{r}_s, \boldsymbol{C}_s\}_{s=1}^t$. Define the martingale difference sequence $X_s = n_i(s) - n_i(s-1) - \hat{\eta}_i(s)$. By the Azuma–Hoeffding inequality, for all $t \in [T]$, we have

$$\mathbb{P}\left(\sum_{s=1}^t n_i(s)-n_i(s-1)-\hat{\eta}_i(s) \le -\sqrt{8t\log T}\right) \le \frac{1}{T^4}. \tag{83}$$

This leads to

$$\mathbb{P}\left(n_i(t) \le \sum_{s=1}^t \hat{\eta}_i(s) - \sqrt{8t\log T}, \text{ for some } t \in [T]\right) \le \frac{1}{T^3}. \tag{84}$$

Therefore, we know that $n_i(t) \geq \frac{\chi t}{20}$ holds with probability $1 - \frac{1}{T^3}$ if $\boldsymbol{b}^{(s)} \in [\boldsymbol{b} - \epsilon, \boldsymbol{b} + \epsilon]$ for all $s \in [t]$ and $t \geq \frac{3200 \log T}{\chi^2}$, and the bias in the resource consumption can be upper bounded as

$$\begin{aligned}
\|\mathbb{E}\left[\boldsymbol{C}_{\cdot,i_{t+1}}(\boldsymbol{b}^{(t)})|\mathcal{H}_t\right] - \boldsymbol{b}^{(t)}\|_\infty &= \|\boldsymbol{C}\hat{\boldsymbol{\eta}}(t) - \boldsymbol{b}^{(t)}\|_\infty \\
&\leq \|\boldsymbol{C}\tilde{\boldsymbol{\eta}}(t) - \boldsymbol{b}^{(t)}\|_\infty + \|\boldsymbol{C}\|_\infty \|(\tilde{\boldsymbol{\eta}}(t) - \hat{\boldsymbol{\eta}}(t))\|_\infty \\
&\leq \| \left(\boldsymbol{C} - \boldsymbol{C}^L(t)\right) \left(\boldsymbol{C}^L(t)\right)^{-1} \boldsymbol{b}^{(t)}\|_\infty + \frac{4 \min\{m^{\frac{5}{2}}, d^{\frac{5}{2}}\}\epsilon_{\mathrm{LP}}}{\sigma \log^2 T} \\
&\leq \|\boldsymbol{C} - \boldsymbol{C}^L(t)\|_\infty \| \left(\boldsymbol{C}^L(t)\right)^{-1} \|_\infty \|\boldsymbol{b}^{(t)}\|_\infty + \frac{4 \min\{m^{\frac{5}{2}}, d^{\frac{5}{2}}\}\epsilon_{\mathrm{LP}}}{\sigma \log^2 T} \\
&\leq \min\{m, d\} \sqrt{\frac{40 \log T}{\chi t}} \cdot \frac{2\sqrt{d}}{\sigma} \cdot \frac{6b}{5} + \frac{4 \min\{m^{\frac{5}{2}}, d^{\frac{5}{2}}\}\epsilon_{\mathrm{LP}}}{\sigma \log^2 T} \\
&\leq \frac{16 b \min\{m, d\}\sqrt{d}}{\sigma} \sqrt{\frac{\log T}{t}} + \frac{4 \min\{m^{\frac{5}{2}}, d^{\frac{5}{2}}\}\epsilon_{\mathrm{LP}}}{\sigma \log^2 T}.
\end{aligned} \tag{85}$$

The last line is defined to be ϵ_t when $t > \frac{\alpha T}{19}$. Otherwise, when $t \leq \frac{\alpha T}{19}$, ϵ_t is defined to be $\frac{6}{5}$. Given α, consider the minimum $\underline{T}$ that satisfies $\frac{\alpha T}{19} > \frac{3200 \log T}{\chi^2}$. For any $T > \underline{dT}$, we have the following two cases:

1. $t \geq \frac{\alpha T}{19}$. In this case, we upper bound the gap between the expected resource consumption and the average remaining resource as shown in (85). From the definition of ϵ_t, we have $\|\mathbb{E}\left[\boldsymbol{C}_{\cdot,i_{t+1}}(\boldsymbol{b}^{(t)})|\mathcal{H}_t\right] - \boldsymbol{b}^{(t)}\|_\infty \leq \epsilon_t$, which implies that the event $\mathcal{E}_t$ is contained in the complement of the following set:
$$\mathcal{E}_t \subset \left\{ n_i(t) \geq \sum_{s=1}^{t} \hat{\eta}_i(s) - \sqrt{8t \log T}, \text{ for all } i \in [m] \right\}^{\complement}. \tag{86}$$
2. $t < \frac{\alpha T}{19}$. Since the resource consumption matrix is element-wise bounded in $[0, 1]$, we have $\mathbf{0} \leq \mathbb{E}\left[\boldsymbol{C}_{\cdot,i_{t+1}}(\boldsymbol{b}^{(t)})|\mathcal{H}_t\right] \leq \mathbf{1}$. In addition, $\mathbf{0} \leq \boldsymbol{b}^{(t)} \leq \frac{\mathbf{6}}{\mathbf{5}}$. As a result, $\|\mathbb{E}\left[\boldsymbol{C}_{\cdot,i_{t+1}}(\boldsymbol{b}^{(t)})|\mathcal{H}_t\right] - \boldsymbol{b}^{(t)}\|_\infty \leq \frac{6}{5} = \epsilon_t$. Since the event $\mathcal{E}_t$ requires $\|\mathbb{E}\left[\boldsymbol{C}_{\cdot,i_{t+1}}(\boldsymbol{b}^{(t)})|\mathcal{H}_t\right] - \boldsymbol{b}^{(t)}\|_\infty$ to be greater than ϵ_t, it follows that $\mathcal{E}_t = \emptyset$ in this case.

Combining the two cases, we obtain the following bound, which completes the proof of Lemma 7:

$$\mathbb{P}\left(\bigcup_{t=1}^{T} \mathcal{E}_t\right) \subset \mathbb{P}\left(\left\{ n_i(t) \geq \sum_{s=1}^{t} \hat{\eta}_i(s) - \sqrt{8t \log T}, \text{ for all } t \in [T] \text{ and all } i \in [m] \right\}^{\complement}\right) \leq \frac{\min\{m, d\}}{T^3}. \tag{87}$$

Proof of Lemma 8 In the proof of Lemma 8, our argument deviates from that of Li et al. (2021) by employing a different choice of ϵ_t, which reflects the approximation error incurred when solving the LP problems. For any $j \in [d]$, define

$$\tau_j' = \min\{t : b_j^{(s)} \notin [b - \epsilon, b + \epsilon]\}. \tag{88}$$

That is, τ_j' represents the first time the j-th entry of $\boldsymbol{b}^{(s)}$ exceeds the interval $[\boldsymbol{b} - \epsilon, \boldsymbol{b} + \epsilon]$. Recall that $\tau' = \min\{t : \boldsymbol{b}^{(t)} \notin [\boldsymbol{b} - \epsilon, \boldsymbol{b} + \epsilon]\}$, which implies that $\tau' = \min_j\{\tau_j'\}$. Consequently, the following event holds:

$$\left\{\tau' \leq t, \bigcap_{s=1}^{t} \mathcal{E}_s^{\complement}\right\} = \bigcup_{j \in [d]} \left\{\tau_j' \leq t, \ \tau_k' \geq \tau_j' \text{ for } k \in [j] \text{ and } k \neq j \ , \bigcap_{s=1}^{t} \mathcal{E}_s^{\complement}\right\}. \tag{89}$$

With $\mathcal{F}_j = \left\{\tau_k' \geq \tau_j' \text{ for } k \in [j] \text{ and } k \neq j\right\}$, we obtain the following probability bound:

$$\mathbb{P}\left(\tau' \leq t, \bigcap_{s=1}^{t} \mathcal{E}_s^{\complement}\right) \leq \sum_{j \in [d]} \mathbb{P}\left(\tau_j' \leq t, \ \mathcal{F}_j, \bigcap_{s=1}^{t} \mathcal{E}_s^{\complement}\right). \tag{90}$$

Next, we derive a bound for the probability corresponding to each constraint $j \in [d]$. Given the original sequence $b_j^{(s)}$ We define the sequence $\tilde{b}_j^{(s)}$ as follows:

$$\tilde{b}_j^{(s)} = \begin{cases} b_j^{(s)} & s \le \tau_j' \\ \tilde{b}_j^{(s-1)} & s > \tau_j' \end{cases}. \tag{91}$$

It is straightforward to see that $\tilde{b}_j^{(s)}$ can be interpreted as a truncated version of $b_j^{(s)}$ when it exceeds the threshold ϵ. Notably, we have the following equivalence:

$$\left\{ b_j^{(s)} \notin [b-\epsilon, b+\epsilon] \text{ for some } s \le t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t} \mathcal{E}_t^{\complement} \right\} = \left\{ \tilde{b}_j^{(s)} \notin [b-\epsilon, b+\epsilon] \text{ for some } s \le t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t} \mathcal{E}_t^{\complement} \right\}. \tag{92}$$

The definition of $\tilde{b}_j^{(s)}$ has the following properties. Consider a sample point

$$\omega = \{\tilde{b}_j^{(0)}, \tilde{b}_j^{(1)}, \ldots, \tilde{b}_j^{(t)}\} \in \left\{ \tilde{b}_j^{(s)} \notin [b-\epsilon, b+\epsilon] \text{ for some } s \le t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t} \mathcal{E}_t^{\complement} \right\}, \tag{93}$$

where ω is a sequence satisfying

1. $\tilde{b}_j^{(s)} \in [b-\epsilon, b+\epsilon]$, for $s < \tau_j'$,
2. $\tilde{b}_j^{(s)} \equiv \tilde{b}_j^{(\tau_j')} \notin [b-\epsilon, b+\epsilon]$, for $\tau_j' \le s \le t$,
3. $\tilde{b}_k^{(s)} \in [b-\epsilon, b+\epsilon]$ for $s \le \tau_j'$ and $k \in [d],\ k \ne j$.

Then from the definition of $\tilde{b}_j^{(s)}$ and $b_j^{(s)}$, the increment $\Delta_s = \tilde{b}_j^{(s)} - \tilde{b}_j^{(s-1)}$ is given by

$$\Delta_s = \tilde{b}_j^{(s)} - \tilde{b}_j^{(s-1)} = \begin{cases} 0, & s > \tau_j', \\ -\frac{1}{T-s}(C_{j,s} - \tilde{b}_j^{(s-1)}), & s \le \tau_j'. \end{cases} \tag{94}$$

With $\tilde{b}_j^{(0)} = b$, we obtain

$$\tilde{b}_j^{(s)} - b = \sum_{k=1}^{s} \Delta_k = \sum_{k=1}^{s} \mathbb{E}[\Delta_k | \mathcal{H}_{k-1}] + \sum_{k=1}^{s} (\Delta_k - \mathbb{E}[\Delta_k | \mathcal{H}_{k-1}]). \tag{95}$$

From the definition of $\bigcap_{s=1}^{t} \mathcal{E}_t^{\complement}$ and the properties of the sequence $\tilde{b}_j^{(s)}$, we observe that:

1. Since $\{\tilde{b}_j^{(s)} \in [b-\epsilon, b+\epsilon], \text{for } s < \tau_j'\}$ and $\{\tau_k' \ge \tau_j' \text{ for } k \in [j] \text{ and } k \ne j\}$, we have $\mathbb{E}[\Delta_s | \mathcal{H}_{s-1}] \le \frac{\epsilon_s}{T-s}$, for $s \le \tau_j'$.
2. Since $\{\tilde{b}_j^{(s)} \equiv \tilde{b}_j^{(s-1)}, \text{for } s > \tau_j'\}$, it follows that $\mathbb{E}[\Delta_s | \mathcal{H}_{s-1}] \equiv 0$ for $s > \tau_j'$.

For the first summation in (95), it is bound by

$$\left| \sum_{k=1}^{s} \mathbb{E}[\Delta_k | \mathcal{H}_{k-1}] \right| \le \sum_{k=1}^{T} |\mathbb{E}[\Delta_k | \mathcal{H}_{k-1}]| = \sum_{k=1}^{\alpha T} \frac{\bar{\epsilon}}{T-k} + \sum_{k=\alpha T+1}^{T-1} \frac{\epsilon_k}{T-k} \le \frac{\alpha \bar{\epsilon}}{1-\alpha} + \sum_{k=\alpha T+1}^{T-1} \frac{\epsilon_k}{T-k}. \tag{96}$$

Consequently, if condition equation 71 holds, namely, $\frac{\alpha\bar{\epsilon}}{1-\alpha} + \sum_{t=\alpha T+1}^{T-1} \frac{\epsilon_t}{T-t} \le \frac{2\epsilon}{3}$, we have $|\sum_{k=1}^{s} \mathbb{E}[\Delta_k | \mathcal{H}_{k-1}]| \le \frac{2\epsilon}{3}$.

For the second summation in (95), the term $\Delta_k - \mathbb{E}[\Delta_k | \mathcal{H}_{k-1}]$ forms a martingale difference sequence. Moreover, since $C_{j,t} \in [0,1]$ and $\mathbb{E}[C_{j,t} | \mathcal{H}_{k-1}] \in [0,1]$, we obtain the bound

$$|\Delta_k - \mathbb{E}[\Delta_k | \mathcal{H}_{k-1}]| = \frac{1}{T-k} \left| C_{j,t} - \tilde{b}_j^{(t-1)} - \left(\mathbb{E}[C_{j,t} | \mathcal{H}_{k-1}] - \tilde{b}_j^{(t-1)} \right) \right| \le \frac{1}{T-k}. \tag{97}$$

Applying the Azuma–Hoeffding inequality, we derive

$$\mathbb{P}\left(\left|\sum_{k=1}^{s}\Delta_k - \mathbb{E}[\Delta_k|\mathcal{H}_{k-1}]\right| \geq \frac{\epsilon}{3}, \text{for some } s \leq t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_t^{\complement}\right) \leq 2\exp\left(-\frac{(\frac{\epsilon}{3})^2}{2\sum_{k=1}^{t}\frac{1}{(T-k)^2}}\right). \tag{98}$$

To further simplify the probability, we note that $\sum_{k=1}^{t}\frac{1}{(T-k)^2} \leq \sum_{k=1}^{t}\frac{1}{(T-k)(T-k-1)} \leq \frac{1}{T-t-1}$. Substituting this result, we obtain

$$\mathbb{P}\left(\left|\sum_{k=1}^{s}\Delta_k - \mathbb{E}[\Delta_k|\mathcal{H}_{k-1}]\right| \geq \frac{\epsilon}{3}, \text{for some } s \leq t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_t^{\complement}\right) \leq 2\exp\left(-\frac{(T-t-1)\epsilon^2}{18}\right). \tag{99}$$

Finally, the event $\left\{\tilde{\tau}_j' \leq t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_t^{\complement}\right\}$ can be appropriately bounded as follows:

$$\begin{aligned}
&\left\{\tilde{\tau}_j' \leq t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_t^{\complement}\right\} &(100)\\
=&\left\{b_j^{(s)} \notin [b-\epsilon, b+\epsilon] \text{ for some } s \leq t, \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_t^{\complement}\right\}\\
\subset&\left\{\left|\sum_{k=1}^{s}\Delta_k\right| \geq \epsilon \text{ for some } s \leq t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_t^{\complement}\right\}\\
\subset&\left\{\left|\sum_{k=1}^{s}(\Delta_k - \mathbb{E}[\Delta_k|\mathcal{H}_{k-1}])\right| \geq \frac{\epsilon}{3} \text{ for some } s \leq t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_t^{\complement}\right\}\bigcup\left\{\left|\sum_{k=1}^{s}\mathbb{E}[\Delta_k|\mathcal{H}_{k-1}]\right| > \frac{2\epsilon}{3},\ \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_t^{\complement}\right\}\\
=&\left\{\left|\sum_{k=1}^{s}(\Delta_k - \mathbb{E}[\Delta_k|\mathcal{H}_{k-1}])\right| \geq \frac{\epsilon}{3} \text{ for some } s \leq t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_t^{\complement}\right\}, &(101)
\end{aligned}$$

which indicates that

$$\begin{aligned}
\mathbb{P}\left(\tau' \leq t,\ \bigcap_{s=1}^{t}\mathcal{E}_s^{\complement}\right) &\leq \sum_{j\in[d]}\mathbb{P}\left(\tau_j' \leq t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_s^{\complement}\right)\\
&= \sum_{j\in[d]}\mathbb{P}\left(\left|\sum_{k=1}^{s}\Delta_k - \mathbb{E}[\Delta_k|\mathcal{H}_{k-1}]\right| \geq \frac{\epsilon}{3}, \text{for some } s \leq t,\ \mathcal{F}_j,\ \bigcap_{s=1}^{t}\mathcal{E}_t^{\complement}\right)\\
&\leq 2d\exp\left(-\frac{(T-t-1)\epsilon^2}{18}\right). &(102)
\end{aligned}$$