



HAL
open science

A Competitive Posted-Price Mechanism for Online Budget-Feasible Auctions

Andreas Charalampopoulos, Dimitris Fotakis, Panagiotis Patsilinakos, Thanos
Tolias

► **To cite this version:**

Andreas Charalampopoulos, Dimitris Fotakis, Panagiotis Patsilinakos, Thanos Tolias. A Competitive Posted-Price Mechanism for Online Budget-Feasible Auctions. EC '25: 26th ACM Conference on Economics and Computation, Jul 2025, Stanford, United States. <10.1145/3736252.3742668>. <hal-05431126>

HAL Id: hal-05431126

<https://hal.science/hal-05431126v1>

Submitted on 24 Dec 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

A Competitive Posted-Price Mechanism for Online Budget-Feasible Auctions*

Andreas Charalampopoulos^{1,2}, Dimitris Fotakis^{1,2}, Panagiotis Patsilinakos³, and Thanos Tolias^{1,2}

¹ School of ECE, National Technical University of Athens, Greece

² Archimedes, Athena Research Center, Greece

³ Université Paris-Dauphine, Université PSL, CNRS, LAMSADE, 75016, Paris, France
andcharalamp@gmail.com, fotakis@cs.ntua.gr, thanostolias@mail.ntua.gr,
panagiotis.patsilinakos@dauphine.psl.eu

Abstract. We consider online procurement auctions, where the agents arrive sequentially, in random order, and have private costs for their services. The buyer aims to maximize a monotone submodular value function for the subset of agents whose services are procured, subject to a budget constraint on their payments. We consider a posted-price setting where upon each agent’s arrival, the buyer decides on a payment offered to them. The agent accepts or rejects the offer, depending on whether the payment exceeds their cost, without revealing any other information about their private costs whatsoever. We present a randomized online posted-price mechanism with constant competitive ratio, thus resolving the main open question of (Badanidiyuru, Kleinberg and Singer, EC 2012). Posted-price mechanisms for online procurement typically operate by learning an estimation of the optimal value, denoted as OPT, and using it to determine the payments offered to the agents. The main challenge is to learn OPT within a constant factor from the agents’ accept / reject responses to the payments offered. Our approach is based on an online test of whether our estimation is too low compared against OPT and a carefully designed adaptive search that gradually refines our estimation.

1 Introduction

We consider a model of procurement auctions, introduced by Singer [1], where a set \mathcal{N} of n strategic agents are willing to offer their services to a buyer. Each agent $i \in \mathcal{N}$ has a private cost $c_i \in \mathbb{R}_{\geq 0}$ for offering their service. The buyer aims to maximize a public monotone submodular value function $f : 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ for the subset of agents hired, subject to a hard budget constraint that the total amount of payments to the agents should not exceed the buyer’s budget B .

In the algorithmic problem, the buyer is aware of the agent costs c_1, \dots, c_n and aims to compute a subset $S \subseteq \mathcal{N}$ of agents (a.k.a. a solution) that maximizes $f(S)$ subject to $\sum_{i \in S} c_i \leq B$. Maximizing a monotone submodular function subject to a budget (a.k.a. knapsack) constraint is a classical NP-hard optimization problem whose approximability is well understood (see e.g., [2,3,4,5]).

In the mechanism design setting of [1,6], we aim to design *individually rational* and *truthful* (a.k.a. *incentive compatible*) mechanisms that compensate each agent i included in the solution with a carefully chosen payment p_i . Individual rationality requires that $p_i \geq c_i$, i.e., that the payment allocated to each agent in the solution should be no less than their cost. Truthfulness requires that the agent selection and the corresponding payments should ensure that no agent has an incentive to falsely report a higher cost to the mechanism, in an attempt to extract a larger payment. The mechanism must be budget-feasible with respect to the payments p_i to the agents included in the solution, i.e., it must be $\sum_{i \in S} p_i \leq B$, and in the ideal case, the mechanism should also be computationally efficient. Conceptually, individual rationality and truthfulness require that the payments are large enough to cover the agent costs and incentivize truthful reporting, while budget feasibility requires that the payments are kept low, so that the value of the solution is maximized.

* This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

Compensating strategic agents for their effort in a way that is incentive-compatible, value-efficient and budget-feasible is an intriguing algorithmic problem with significant practical applications, which include crowdsourcing [7,8,9], crowdsensing [10], participatory sensing [11] (see also the references in the survey of Liu et al. [12]).

Since its introduction by Singer [1] and due to its practical and theoretical significance, the problem of designing truthful mechanisms for budget-feasible procurement auctions has received significant attention. A long line of research has shown that budget-feasible procurement can be approximated within small constant factors by computationally efficient truthful mechanisms for additive [13], monotone submodular [14,15] and non-monotone submodular valuations [16,15] (see Section 1.2 for a more detailed discussion), in the *bidding* (a.k.a. *sealed-bid*) setting, where the mechanism may ask the agents to report their costs.

Previous work also considered *online* budget-feasible procurement auctions with *secretary arrivals*, where the agents are chosen adversarially and arrive sequentially in random order, and the mechanism has to decide irrevocably about selecting an agent in the solution and the corresponding payment, upon the agent’s arrival and without any knowledge about future agent arrivals. With agent bidding, online procurement auctions admit constant factor approximations for additive [7], monotone submodular [17] and non-monotone submodular [16] valuations.

However, in practical applications, such as crowdsourcing, in addition to dealing with the agents online, it is desirable (and often necessary) that the mechanism interacts with the agents through *sequential posted pricing*. A posted-price mechanism decides on the payment p_i offered to an agent i upon i ’s arrival. Then, the agent accepts or rejects the offer, depending on whether $p_i \geq c_i$ or not, without revealing any other information about their private cost c_i whatsoever. Posted-price mechanisms are used extensively in practice, due to their simplicity, transparency, obvious truthfulness [18] and agent privacy regarding their costs. E.g., posted pricing is the standard form of pricing in crowdsourcing applications, such as Mechanical Turk (see also the discussion in [19,20] for the motivation and the advantages of sequential posted-price mechanisms).

In the context of online budget-feasible procurement, Badanidiyuru, Kleinberg and Singer [17] asked whether *posted-price mechanisms are as powerful, in terms of the asymptotic behavior of their competitive ratio, as standard bidding mechanisms*. Towards resolving this question, they presented a randomized constant-competitive posted-price mechanism for symmetric submodular valuations⁴ and agents sampled independently from an unknown distribution. For monotone submodular valuations and secretary arrivals, Badanidiyuru et al. [17] presented a randomized constant-competitive online mechanism with agent bidding, and a randomized $O(\log n)$ -competitive posted-price mechanism. Their main open question concerned the existence of a constant-competitive posted-price mechanism for monotone submodular valuations and secretary agent arrivals.

Subsequently, Balkanski and Hartline [19] considered posted-price mechanisms in the Bayesian setting, where the agents arrive as independent samples from a known distribution. They presented mechanisms that are constant-competitive against the Bayesian optimal mechanism for additive, symmetric submodular and monotone submodular value functions.

1.1 Contribution and General Approach

In this work, we present a randomized constant-competitive posted-price mechanism for online procurement auctions with monotone submodular valuations, thus resolving the main open question of Badanidiyuru et al. [17]. More formally, we show the following:

⁴ A function $f : 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ is *symmetric submodular* if there is a nondecreasing concave function $g : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ such that $f(S) = g(|S|)$ for all $S \subseteq \mathcal{N}$.

Theorem 1. *There is a universally truthful $O(1)$ -competitive randomized posted-price mechanism for budget-feasible online procurement auctions with secretary agent arrivals and monotone submodular buyer’s valuations.*

The mechanism of Theorem 1 satisfies budget-feasibility with certainty, and is individually rational and universally (and obviously) truthful⁵, with the latter properties following directly from the definition of posted-price mechanisms, where the agents to act in their best interest when selecting whether to accept the payment offered or not.

Posted-price mechanisms for online procurement typically operate by learning an estimation of the optimal value OPT (usually referred to as a *threshold*, for brevity) \hat{t} and using it to determine the payments (a.k.a. *prices*) offered to the agents. We restrict our attention to *linear-price* mechanisms, where the price offered to agent i is $p_i = f_S(i)B/\hat{t}$, where B is the buyer’s budget, \hat{t} is the mechanism’s current threshold, S is the subset of agents selected in the solution up to i ’s arrival, and $f_S(i) = f(S \cup \{i\}) - f(S)$ is the marginal value due to i ’s inclusion in the solution.

One can show (see e.g., the proof of Lemma 12 or [17, Lemma 4.4]) that linear-price mechanisms with thresholds \hat{t} that remain within a constant fraction of OPT for a constant fraction of the agent sequence are constant-competitive. Hence, the main challenge in the design of efficient linear-price mechanisms (and the main difference between them) is how to maintain a threshold \hat{t} that mostly remains within a constant fraction of OPT.

The posted-price mechanism of Badanidiyuru et al. [17, Section 4] uses linear prices and a fixed threshold \hat{t} determined before the first offer to an agent. In Theorem 2, we show that the competitive ratio of any linear-price mechanism that selects a fixed threshold without any knowledge of the agent costs is $\Omega(\log n)$. Hence, we consider *adaptive linear-price* mechanisms based on a threshold that evolves over time.

Our approach is inspired by the constant-competitive adaptive linear-price mechanism of [17, Section 3] for symmetric submodular valuations and agents sampled independently from an unknown distribution. As in [17, Section 3], our approach is based on an online test (cf. Mechanism 2, usually referred to as *TestThreshold*) that efficiently decides if the current threshold \hat{t} is less than $\text{OPT}/4$. TestThreshold comes with an one-sided guarantee: with constant probability, TestThreshold succeeds (i.e., it responds that the current threshold $\hat{t} \leq \text{OPT}/4$), if it is applied with any threshold $\hat{t} \leq \text{OPT}/4$ to a random subset of agents of size $\Omega(v_{\max}n/\text{OPT})$, where $v_{\max} = \max_{i \in \mathcal{N}}\{f(\{i\})\}$ is the maximum value of an agent (Property 2). If TestThreshold is applied with a threshold $\hat{t} > \text{OPT}/4$, there is no guarantee on its response. If TestThreshold fails (i.e., it responds that the current threshold $\hat{t} > \text{OPT}/4$), we become aware that \hat{t} is rather too large and should be decreased; otherwise, TestThreshold contributes $\Omega(v_{\max}\hat{t}/\text{OPT})$ to the total value of the mechanism.

The high level idea of our approach is to first obtain a rough estimation of OPT and then apply binary search, guided by TestThreshold, to the possible range of OPT defined by our initial estimation. At the conceptual level, our approach consists of three periods:

Learning v_{\max} : As in standard secretary algorithms, we start with a *learning period* that considers a constant fraction of the agents, where we do not make any offers and aim to estimate v_{\max} with constant probability. We note that $v_{\max} \leq \text{OPT} \leq n \cdot v_{\max}$.

Binary Search: Using TestThreshold, we apply binary search to the set $\{v_{\max}, 2v_{\max}, 4v_{\max}, \dots, 2^{\log n}v_{\max}\}$ of different estimations of OPT expressed as a power of 2 times v_{\max} . This period aims to compute a threshold \hat{t} within a constant fraction of OPT. Our application of binary

⁵ A randomized mechanism is universally truthful if it can be expressed as a probability distribution over truthful deterministic mechanisms. A mechanism is *obviously truthful* [18] if it has an equilibrium in obviously dominant strategies. A strategy is *obviously dominant* if the best outcome under any possible deviation from it is no better than the worst outcome under the given strategy.

search requires $\Theta(\log \log n)$ *phases*, each consisting of $O(\log \log n)$ negatively-dependent *rounds* where we apply TestThreshold (Appendix B.4). Hence, we can prove that using $O((\log \log n)^2)$ applications of TestThreshold to the next constant fraction of agents, we end up with a threshold $\hat{t} \geq \text{OPT}/8$ with constant probability (Lemma 9).

Exploitation: Applying TestThreshold to the last constant fraction of agents, where every time the test succeeds \hat{t} is doubled, and every time the test fails \hat{t} is halved, we collect an expected total value that is within a constant fraction of OPT (Lemma 10).

The three-period approach above comprises a constant-competitive posted-price mechanism for online budget-feasible procurement, but only under a *large market assumption*. Specifically, in order to show a constant competitive ratio, we need to assume that $\text{OPT}/v_{\max} = \Omega((\log \log n)^2)$, i.e., that the optimal solution consists of sufficiently many agents. Such large market is necessary in order to guarantee that each of the $O((\log \log n)^2)$ applications of TestThreshold considers a random subset of agents of size $\Omega(v_{\max}n/\text{OPT})$. This is necessary, so that we can prove that the entire period of Binary Search succeeds with constant probability.

We can deal with the case where $\text{OPT}/v_{\max} = o((\log \log n)^2)$ by applying Dynkin’s secretary algorithm [21] with constant probability, which results in competitive ratio of $O((\log \log n)^2)$ (which can be improved to $O(\log \log n)$ with a careful analysis).

To remove the large market assumption and obtain a constant-competitive mechanism, we assume that $\text{OPT} \geq \beta v_{\max}$, where β is a sufficiently large constant. We define a power tower sequence as $\psi_1 = \beta$ and $\psi_{j+1} = 2^{\psi_j} \cdot \psi_j$ and apply TestThreshold with thresholds $\psi_j v_{\max}$ in order to determine a pair of consecutive intervals $[\psi_j v_{\max}, \psi_{j+1} v_{\max})$ and $[\psi_{j+1} v_{\max}, \psi_{j+2} v_{\max})$ such that OPT is included in their union. We show how to compute such a pair of intervals with constant probability by $O(\log^* n)$ applications of TestThreshold to a constant fraction of the agent sequence in total (Lemma 20). Selecting one of these intervals $[\psi_j v_{\max}, \psi_{j+1} v_{\max})$ and $[\psi_{j+1} v_{\max}, \psi_{j+2} v_{\max})$ at random and applying Binary Search and Exploitation, as described above, to the chosen interval (instead of the interval $[\beta v_{\max}, nv_{\max}]$ initially considered) satisfies the required large market property and results in a constant-competitive posted-price mechanism, as required by Theorem 1.

1.2 Related Work

From an algorithmic viewpoint, the greedy algorithm of Nemhauser et al. [2] gives an approximation ratio of $e/(e-1)$ for the algorithmic problem of maximizing a monotone submodular function subject to a knapsack constraint [4], and this approximation guarantee is best possible in polynomial-time [3], under standard complexity assumptions. More recently, Badanidiyuru et al. [5] presented a $(9/8 + \varepsilon)$ -approximation with polynomially many demand queries.

Singer [1,6] was the first to study the approximability of budget-feasible procurement auctions by truthful mechanisms and presented a randomized $O(1)$ -approximation for monotone submodular valuations. Subsequently, Chen et al. [22] significantly improved the approximation ratio to 7.91 (resp. 3) for randomized and 8.34 (resp. $2 + \sqrt{2}$) for deterministic mechanisms when the buyer’s valuation is monotone submodular (resp. additive), using a greedy (resp. knapsack) based strategy. Chen et al. also proved an unconditional lower bound of $1 + \sqrt{2}$ (resp. 2) on the approximability of budget-feasible auctions with additive values by deterministic truthful (resp. randomized universally truthful) mechanisms. For additive values, Gravin et al. [13] matched the best possible approximation ratio of 2 for randomized mechanisms and presented a 3-approximate deterministic mechanism. Anari et al. [23] gave best possible $e/(e-1)$ -approximation mechanisms for large markets.

For monotone submodular valuations, Jalaly and Tardos [14] presented a randomized polynomial-time 5-approximation mechanism, thus improving on [22], and a deterministic (resp. randomized)

4.56 (resp. 4) approximation mechanism, which assumes access to an exact algorithm for value maximization. The approximation ratio for large markets was improved to 2 for deterministic (possibly exponential-time) mechanisms and to 3 for randomized polynomial-time mechanisms by Anari et al. [23]. Recently, Balkanski et al. [15] presented a deterministic polynomial-time clock auction that is 4.75-approximate for monotone submodular valuations. The best known approximation ratio for monotone submodular valuations is 4.45 for deterministic and 4.3 for randomized mechanisms, achieved by the polynomial-time clock auction of Han et al. [24].

Polynomial-time constant-factor approximations are also known for non-monotone submodular valuations [16,25], where the best known ratio is 64 for deterministic [15] and 12 for randomized mechanisms [24] (both achieved by clock auctions). Moreover, constant-factor approximation mechanisms (albeit non-polynomial time ones) are known for XOS valuations [26,27], while for subadditive valuations the best known approximation guarantee is $O(\frac{\log n}{\log \log n})$ due to Balkanski et al. [15], which is achieved by a clock auction, matches the randomized approximation of [26] and improves on the $O(\log^3 n)$ deterministic approximation of [28]. We refer an interested reader to the survey of Liu et al. [12] for a detailed discussion of previous work on budget-feasible procurement.

The online version of budget-feasible auctions, where the agents arrive sequentially in random order and the decision about their acceptance in the solution is online and irrevocable, was introduced by Singer and Mittal [7] and Badanidiyuru et al. [17]. Amanatidis et al. [16] presented $O(1)$ -competitive randomized online universally truthful mechanisms for monotone and non-monotone submodular valuations in the bidding setting. Online budget-feasible procurement is closely related to the *submodular knapsack secretary* problem [29,30], where budget feasibility is with respect to the agent costs (which are revealed to the algorithm upon each agent's arrival). Feldman et al. [31] presented a randomized $20e$ -approximation for submodular knapsack secretaries.

The clock auctions of Balkanski et al. [15] and Han et al. [24] achieve small constant approximation guarantees without resorting to bidding (i.e., the agents never report their costs to the mechanism), but they are not online, because the agents receive multiple offers by the mechanism. To the best of our knowledge, the online $O(\log n)$ -competitive mechanism of [17, Section 4] and the constant-competitive mechanisms in the Bayesian setting of [19] are the only known posted-price mechanisms for online budget-feasible procurement auctions with monotone submodular valuations.

A quite standard approach to the design of efficient budget-feasible mechanisms (see e.g., [17, Section 5] and in [16, Section 4]) and of online algorithms for submodular knapsack secretaries (see e.g., [31]) is to first learn the costs of a random subset of agents. Then, based on these costs, the mechanism approximates the optimal solution of the resulting random instance and uses its value as a threshold to post linear prices to the remaining agents. We should highlight that this approach does not fit in the framework of online posted-price mechanisms, since it requires knowledge of the costs of a significant fraction of agents (which is obtained through bidding).

2 Model, Definitions and Preliminaries

A set function $f : 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ is *non-decreasing* (often referred to as *monotone*) if for all $S \subseteq T \subseteq \mathcal{N}$, $f(S) \leq f(T)$. A function f is *submodular* if f has non-increasing marginal values, i.e., for all $S \subseteq T \subseteq \mathcal{N}$ and all $i \notin T$, $f(T \cup \{i\}) - f(T) \leq f(S \cup \{i\}) - f(S)$. We let $f_S(i) = f(S \cup \{i\}) - f(S)$ denote the marginal value of i with respect to S . In this work, we consider valuation functions $f : 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ that are normalized, i.e. have $f(\emptyset) = 0$, monotone and submodular.

A valuation function f is accessed by *value queries*, which for any given set $S \subseteq \mathcal{N}$, return $f(S)$. We let $v_{\max} = \max_{i \in \mathcal{N}} \{f(\{i\})\}$ denote the maximum value of any agent. In the following, all logarithms are base-2 unless stated otherwise.

Budget-Feasible Procurement Auctions and Mechanisms. We consider procurement auctions with a set $\mathcal{N} = \{1, \dots, n\}$ of agents. Each agent i has a private cost $c_i \in \mathbb{R}_{\geq 0}$ for participating in the solution. The buyer has a budget $B \in \mathbb{R}_{\geq 0}$ and aims to maximize a monotone submodular valuation $f : 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$, subject to the constraint that the sum of payments to the agents must be at most B . A (deterministic) *direct revelation mechanism* $\mathcal{M} = (A, p)$ is a pair consisting of an allocation function and a payment function. The allocation function $A : \mathbb{R}_{\geq 0}^n \rightarrow 2^{\mathcal{N}}$ maps a bid vector $\mathbf{b} = (b_1, \dots, b_n)$ submitted by the agents to a solution $A(\mathbf{b}) = S \subseteq \mathcal{N}$. The payment function $p : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}_{\geq 0}^n$ computes the payments $p(\mathbf{b})$ allocated to the agents for the solution $A(\mathbf{b})$. A randomized mechanism is a probability distribution over deterministic mechanisms.

Given a mechanism $\mathcal{M} = (A, p)$, each agent i aims to maximize their *utility* $u_i(\mathbf{b})$ through their bid. It is $u_i(\mathbf{b}) = p_i(\mathbf{b}) - c_i$, if i is included in the solution $A(\mathbf{b})$, and $u_i(\mathbf{b}) = p_i(\mathbf{b})$, otherwise.

A mechanism is *budget-feasible* if for every bid vector \mathbf{b} , $\sum_{i \in \mathcal{N}} p_i(\mathbf{b}) \leq B$. A mechanism is *individually rational*, if for every bid vector \mathbf{b} and all agents $i \in \mathcal{N}$, $u_i(\mathbf{b}) \geq 0$. Hence, $p_i(\mathbf{b}) \geq 0$ for all agents $i \in \mathcal{N}$, and $p_i(\mathbf{b}) \geq c_i$ for all agents i included in the solution $A(\mathbf{b})$. Due to individual rationality and the budget constraint, we always let $p_i(\mathbf{b}) = 0$ for all agents $i \notin A(\mathbf{b})$.

A mechanism is *truthful*, if reporting their cost c_i is a dominant strategy for every agent i , i.e., for all $i \in \mathcal{N}$, all $b \in \mathbb{R}_{\geq 0}$ and every bidding vector \mathbf{b}_{-i} of the other agents, $u_i(\mathbf{b}_{-i}, c_i) \geq u_i(\mathbf{b}_{-i}, b)$.

For randomized mechanisms, that we consider in this work, we require that the mechanism is budget feasible and individually rational *with certainty* and *universally truthful*, i.e., the mechanism is a probability distribution over deterministic truthful mechanisms.

Online Budget-Feasible Procurement Auctions. In the online setting, agents arrive sequentially and the decision about whether the present agent i is included in the solution and i 's payment is *irrevocable* and is taken upon i 's arrival, using only information about the agents having arrived before i and without any knowledge about future agent arrivals whatsoever.

In this work, we consider an online procurement auctions with *secretary arrivals*, where the set \mathcal{N} of agents is chosen adversarially and the agents arrive in random order (formally, the agents' arrival order is drawn uniformly at random from all permutations of \mathcal{N}).

Posted-Price Mechanisms. We restrict our attention to (online) sequential *posted-price* mechanisms. A posted-price mechanism decides on the price p_i of each agent i upon i 's arrival and makes i a take-it-or-leave-it offer. Agent i accepts or rejects the offer, depending on whether $p_i \geq c_i$ or not, without revealing any other information about their private cost c_i whatsoever. If agent i accepts the offer, i is included in the mechanism's solution and the budget currently available decreases by p_i . Otherwise, we let i 's price be $p_i = 0$ and agent i is discarded.

Randomized posted-price mechanisms are individually rational and universally (and obviously) truthful, because among the two options available (accept or reject the mechanism's offer), the agents select the option that maximizes their utility $\max\{0, c_i - p_i\}$. Moreover, posted-price mechanisms are budget feasible, assuming that each price offered does not exceed the budget currently available.

We further restrict our attention to *linear-price* mechanisms, which maintain a threshold \hat{t} and offer a price $p_i = f_S(i)B/\hat{t}$ to every agent i (assuming that p_i does not exceed the budget currently available). The threshold \hat{t} is an estimation of the optimal value and is maintained based on the marginal values of previous agents and their accept / reject decisions.

We say that a linear-price mechanism is *adaptive*, if it may use a different threshold for determining the linear price offered to each agent, and *non-adaptive*, if the mechanism decides on a fixed threshold before its first offer to an agent and uses it for all remaining agents.

Competitive Ratio. We evaluate the performance of our online mechanisms using the competitive ratio [32]. We compare the performance our our mechanism against the offline optimum, denoted OPT, which is the value of an optimal solution $S^* \subseteq \mathcal{N}$ that maximized $f(S^*)$ subject to $\sum_{i \in S^*} c_i \leq B$.

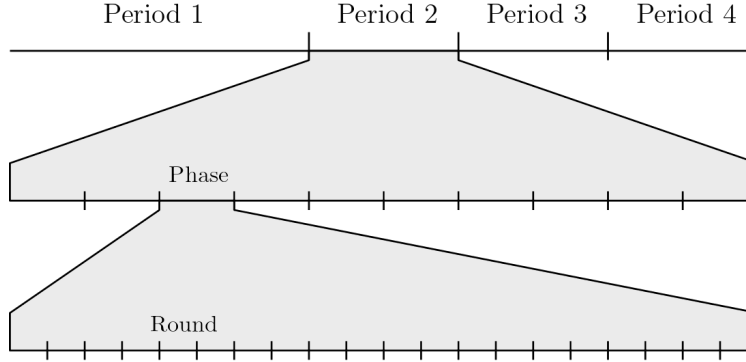


Fig. 1: The input sequence is partitioned into four periods. Periods 2 to 4 are further partitioned into phases, with each phase consisting of multiple rounds.

We note that $\text{OPT} = f(S^*)$ is defined with respect to a computationally unrestricted algorithm with full advance knowledge of the set \mathcal{N} of agents and their costs c_1, \dots, c_n .

In our setting, the competitive ratio cr of an algorithm is the worst-case ratio, over all possible sets of agents \mathcal{N} , of OPT divided by the algorithm's expected cost $\mathbb{E}[\text{ALG}]$ (where the expectation is taken over both the agents' arrival order and the algorithm's random choices).

3 A Logarithmic Lower Bound for Non-Adaptive Linear-Price Mechanisms

To motivate the use of non-adaptive linear-price mechanisms, we first prove the following:

Theorem 2. *Every randomized non-adaptive linear-price mechanism is at least $\frac{\log n}{2}$ -competitive.*

Proof (Proofsketch). We consider a family of instances $\mathcal{I} = \{I_0, I_1, \dots, I_{\log n}\}$. Every instance consists of n value-private cost pairs (v, c) , where for instance I_i , $v = 1$ and $c = B/2^i$, for every $i \in \{0, 1, \dots, \log n\}$.

We construct a probability distribution $\mathcal{D}_{\mathcal{I}}$ over the family \mathcal{I} of instances. For every $i = 0, 1, 2, \dots, \log(n) - 1$, we let I_i occur with probability $p_i = 1/2^{i+1}$ in $\mathcal{D}_{\mathcal{I}}$. We let $I_{\log n}$ occur with probability $p_{\log n} = 1/n$ in $\mathcal{D}_{\mathcal{I}}$, so that the sum of p_i s is equal to 1. To conclude the proof, we show that the expected value of any deterministic non-adaptive linear-price mechanism on $\mathcal{D}_{\mathcal{I}}$ is at most 1, while the expected value of the optimal solution on $\mathcal{D}_{\mathcal{I}}$ is $\frac{\log n}{2} + 1$. Then, the theorem follows from Yao's principle ([32, Chapter 8.4] and [33]). We provide a detailed proof in Section A.

The key idea behind Theorem 2 is that a non-adaptive linear-price mechanism has to decide on a fixed threshold \hat{t} before getting any information about the agents' private costs (which for posted-price mechanisms is obtained only by offering prices to the agents and collecting their accept / reject responses). Hence, essentially the best approach of non-adaptive linear-price mechanisms is to learn v_{\max} and select a threshold \hat{t} from $\{1, 2, 4, \dots, 2^{\log n}\}$ uniformly at random (which is what the posted-price mechanism of [17, Section 4] actually does). Hence, to improve on the logarithmic competitive ratio of [17, Section 4], we next present an adaptive linear-price mechanism.

4 Posted-Price Mechanism: Building Blocks and Outline of the Analysis

We next present the main steps of our posted-price LM-Mechanism and the key ideas of its analysis. LM-Mechanism assumes that OPT/v_{\max} is sufficient large (LM stands for *Large Market*). E.g., it

Algorithm 1: LM-Mechanism

- 1: **Input:** Set \mathcal{N} of agents arriving in random order, budget B
 - 2: **Initialization:** Solution $S \leftarrow \emptyset$ and $\text{val} \leftarrow 0$, budget available $B_{\text{rem}} \leftarrow B$, agents available $\mathcal{N}_{\text{rem}} \leftarrow \mathcal{N}$
 - 3: $v_{\text{max}}, \mathcal{N}_{\text{rem}} \leftarrow \text{LearningMaxValue}(\mathcal{N}_{\text{rem}})$
 - 4: $t_{\text{min}}, t_{\text{max}}, (S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}) \leftarrow \text{PowerTowerSearch}((S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}), B, [v_{\text{max}}, n \cdot v_{\text{max}}])$
 - 5: $t_{\text{init}}, (S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}) \leftarrow \text{BinarySearch}((S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}), B, [t_{\text{min}}, t_{\text{max}}])$
 - 6: $(S, \text{val}) \leftarrow \text{Exploitation}((S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}), B, t_{\text{init}}, [t_{\text{min}}, t_{\text{max}}])$
 - 7: **return** (S, val)
-

suffices that $\text{OPT}/v_{\text{max}} \geq 10^7$. In Section 9, we use Dynkin’s algorithm [21] and an adaptation of the posted-price mechanism in [17, Section 4] to deal with the case where $\text{OPT}/v_{\text{max}} < 10^7$.

As explained in Section 1.1, LM-Mechanism consists of 4 periods. Each of the first three periods “consumes” a constant fraction of the agent sequence (and at most a constant fraction of the budget B) and provides the next period with a more refined estimation of OPT. More specifically, LM-Mechanism proceeds along the following periods:

Period 1 – LearningMaxValue: The first period calls Mechanism 3, usually referred to as LearningMaxValue, which examines about 1/3 of the agent sequence and returns the maximum value of an agent in this part. We let \mathcal{E}_1 denote the event that the value returned by LearningMaxValue is indeed v_{max} , which occurs with probability 1/3. Assuming \mathcal{E}_1 , we have a first rough estimation of OPT, since $v_{\text{max}} \leq \text{OPT} \leq n \cdot v_{\text{max}}$.

Period 2 – PowerTowerSearch: The second period calls Mechanism 4, usually referred to as PowerTowerSearch, which decides on two consecutive intervals $A = [a_{\text{min}}, a_{\text{max}}]$ and $B = [b_{\text{min}}, b_{\text{max}}]$ such that $a_{\text{max}} = 2^{\frac{a_{\text{min}}}{v_{\text{max}}}}$ and $b_{\text{max}} = 2^{\frac{b_{\text{min}}}{v_{\text{max}}}}$. PowerTowerSearch then selects one of the two intervals uniformly at random and returns its endpoints as t_{min} and t_{max} . We let \mathcal{E}_2 denote the event that $\text{OPT} \in A \cup B$ and \mathcal{E}_5 denote the event that conditional on \mathcal{E}_2 , the right interval of A and B is chosen. In Section 6, we discuss the implementation of PowerTowerSearch and prove that \mathcal{E}_2 occurs with probability at least 0.9.

Period 3 – BinarySearch: The third period calls Mechanism 5, usually referred to as BinarySearch, which performs binary search in the interval $[t_{\text{min}}, t_{\text{max}}]$ and returns t_{init} . Assuming a proper execution of BinarySearch, which is implied by event \mathcal{E}_3 formally defined later on, we have that $\text{OPT}/4 \leq t_{\text{init}} \leq O(\log \log(t_{\text{max}}/t_{\text{min}}))\text{OPT}$. The analysis of BinarySearch is presented in Section 7.

Period 4 – Exploitation: The fourth period calls Mechanism 6, usually referred to as Exploitation, which performs an adaptive search, starting with t_{init} , on possible values of the mechanism’s linear-price threshold that are powers of 2 (see also [17, Section 3]). Assuming its proper execution, which is also implied by event \mathcal{E}_3 , Exploitation collects $\Omega(\text{OPT})$ value from its part of the agent sequence. The analysis of Exploitation is presented in Section 8.

Different periods, as they run, update LM-Mechanism’s state with consists of its current solution S , the total value collected so far val , the budget available B_{rem} and the sequence of agents not yet considered \mathcal{N}_{rem} . Each of the last three periods examine a (relatively small) constant fraction of the agent sequence. We let \mathcal{E}_4 denote the event that the total number of agents examined by the 4 periods of LM-Mechanism is at most n (which along with events \mathcal{E}_2 and \mathcal{E}_3 also imply that the total budget expended is at most B). The correctness of LM-Mechanism is implied by event $\mathcal{E} = \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3 \cap \mathcal{E}_4 \cap \mathcal{E}_5$. The correctness probability of LM-Mechanism is analyzed in Section 4.2. The two key technical ingredients in the analysis of LM-Mechanism is a partitioning of the last 3 periods into phases and rounds (as depicted Fig. 1) and TestEstimate, which is applied to each round and tests if the current threshold is too large or too small. We next discuss their main properties.

4.1 Partitioning into Rounds and Phases

To partition the agent sequence into rounds, we employ a standard technique [34], where round lengths are drawn from a binomial distribution to ensure independence among agents and that the resulting prefix of the sequence is a *random subset* of \mathcal{N} , where each agent participates with a given probability (often referred to as the round’s *participation probability* and denoted by q).

Formally, for any suffix $\mathcal{N}' \subseteq \mathcal{N}$ of the agent sequence and any $a \in (0, 1)$, an a -round (or simply a *round*, if a is clear from the context) is a prefix of \mathcal{N}' with length drawn from a binomial distribution $\mathcal{B}(|\mathcal{N}'|, a)$ with parameters $|\mathcal{N}'|$ and a . We usually refer to a as the *length parameter* (or the *length probability*) of the round. Along with the random arrival order of the agents, selecting the length of a round as a binomially distributed random variable ensures that every two agents in \mathcal{N}' are included in the given round independently with probability a (which allows us to analyze the properties of each round using standard concentration inequalities).

Given the agent sequence \mathcal{N} , we define a sequence of $\kappa \geq 1$ rounds with length parameters a_1, \dots, a_κ as follows: we select the length x_1 of the 1st round from the binomial distribution $\mathcal{B}(n, a_1)$, let $n_1 = n - x_1$ be the number of agents remaining, select the length x_2 of the 2nd round from $\mathcal{B}(n_1, a_2)$, \dots , let $n_i = n_{i-1} - x_i$ be the number of agents remaining after the i -th round, select the length x_{i+1} of the $(i + 1)$ -th round from $\mathcal{B}(n_i, a_{i+1})$, \dots , select the length x_κ of the κ -th round from $\mathcal{B}(n_{\kappa-1}, a_\kappa)$. The following analyzes the distribution of the resulting random subsets of agents:

Lemma 1. *For any $\kappa \geq 1$, we consider a sequence of round lengths distributed binomially as described above with length parameters a_1, \dots, a_κ . Then, each agent $b \in \mathcal{N}$ is included in round κ independently of the other agents with participation probability*

$$q_\kappa = a_\kappa \prod_{i=1}^{\kappa-1} (1 - a_i) \tag{1}$$

Proof (Proofsketch). The proof consists of two parts. First we observe that due to secretary agent arrivals, the probability that an agent belongs to any fixed interval of length n_i of an agent sequence of length n is n_i/n . Applying standard properties of the binomial distribution repeatedly, we derive the desired participation probability. Next, we use secretary agent arrivals and the binomial distribution of round lengths and show that the probability that any two agents belong to round κ is q_κ^2 . A detailed proof is given in Appendix B.1.

In our mechanism, we draw multiple rounds with significantly varying lengths, which complicates the calculation of each round’s participation probability using (1). The following lower bound on the participation probability allows us to argue about the properties of each round using its length parameter a . The proof requires the precise definition of the length parameters of all rounds considered by our mechanism and is deferred to Appendix B.2. For the proof, we use that the length parameter of every round is $o(1)$ and that their sum over all rounds is significantly less than 1 (which also allows us to show that event \mathcal{E}_4 occurs with probability at least 0.97, see Lemma 20 in Appendix B.5).

Lemma 2. *The participation probability of any round with length parameter a is at least $6a/(10e)$.*

Good, Dense Rounds and Phases. Our goal is to use TestThreshold on rounds in order to assess whether a threshold t underestimates the true optimum. To this end, we will call an (a, t) -round, a round with length parameter a on which we test a threshold t . For our test to be correct consistently we need the expected number of agents included in both the optimal solution and our (a, t) -round

to be at least a fixed constant, i.e. $a \geq \frac{v_{\max}}{\text{OPT}}$. Since we don't know the value of OPT and t is our estimate of it, we say that an (a, t) -round is 'good' if $a \geq 81 e v_{\max}/t$.

We allocate a budget $B_j = 3C \cdot a_j \cdot B$ to every round j with length parameter a_j , where $C = 1/(8e)$ is a fixed constant used in the following sections. For each round j with set of agents \mathcal{N}_j and length parameter a_j , we let $S_j^* \subseteq \mathcal{N}_j$ be the set of agents in round j with maximum value $f(S_j^*)$ subject to the budget constraint $\sum_{i \in S_j^*} c_i \leq B_j$, and let $\text{OPT}_j = f(S_j^*)$ be the optimal value of round j .

We say that an (a_j, t_j) -round j is *dense* if $\text{OPT}_j \geq C \cdot a_j \cdot \text{OPT}$. Intuitively, a round is dense if it gets a fair share (with respect to its length parameter) of the optimal value. The following shows that 'good' rounds are dense with good probability.

Lemma 3. *A 'good' (a_j, t_j) -round j , such that $t_j \leq \text{OPT}$, is dense with probability at least 0.9.*

The high level intuition behind the proof of Lemma 3 is that a good round is a random set that includes, in expectation, sufficiently many agents of the optimal solution. The requirement on the round being 'good' is mostly technical and is used in the proof of Lemma 3. Lemma 3 is key to the analysis of LM-Mechanism and can be found in Section B.3.

Applying TestEstimate to a dense round with threshold $t \leq \text{OPT}/4$ should verify that t is not too large and can be increased. To amplify the confidence of our test, we apply TestEstimate to a sequence of rounds. A (δ, a, t) -phase is a sequence of δ consecutive rounds all with the same length parameter a and all testing the same threshold t (see also Fig. 1). We note that TestEstimate is always applied with the same threshold t to every round of a given phase. A (δ, a, t) -phase is *dense* if at least $\delta/2$ of its rounds are dense. In Appendix B.4, we show that:

Lemma 4. *A (δ, a, t) -phase, consisting of 'good' rounds and with $t \leq \text{OPT}$, is dense with probability at least $1 - \exp(-\frac{4\delta}{45})$.*

For the proof of Lemma 4, we observe that the events that different a -rounds are dense are negatively associated, in the sense that conditional on a round not being dense, the event that another round is dense becomes more likely. Then, Lemma 4 by an application of standard concentration bounds.

Remark 1. We should highlight that the randomness in the agent arrivals and the internal randomness of LM-Mechanism are only used (i) to partition the agent sequence into rounds consisting of random subsets of agents; and (ii) to select the right interval between A and B in Period 2. Everything else in LM-Mechanism is deterministic. In fact, once the rounds are formed, TestEstimate and the other parts of LM-Mechanism operate without assuming anything about the order in which the agents arrive. \square

4.2 Estimating the Probability of Success

In the following, we always assume that LM-Mechanism gathers positive value only in realizations where event \mathcal{E} holds (and always condition on \mathcal{E} , even without mentioning that). As discussed after the outline of LM-Mechanism, \mathcal{E} is the conjunction of events $\mathcal{E}_1, \dots, \mathcal{E}_5$ formally defined below.

\mathcal{E}_1 : An agent with the maximum utility v_{\max} is included in Period 1.

\mathcal{E}_2 : In Period 2, the interval $[v_{\max}, n \cdot v_{\max}]$ is partitioned into sub-intervals. PowerTowerSearch identifies two consecutive intervals A and B . Then, \mathcal{E}_2 denotes that $\text{OPT}/8 \in A \cup B$.

\mathcal{E}_3 : Every phase during periods 3 and 4 is dense with respect to the threshold \hat{t} used by TestEstimate for that phase.

\mathcal{E}_4 : The sum of the realized round lengths drawn by LM-Mechanism is less than n .

\mathcal{E}_5 : Given \mathcal{E}_2 , the interval I of A and B chosen uniformly at random satisfies $\text{OPT}/8 \in I$.

The following shows that event \mathcal{E} occurs with constant probability:

Lemma 5. *Event \mathcal{E} happens with probability at least $1/20$.*

Proof. In Appendix B.5, we show that $\mathbb{P}[\mathcal{E}_2] \geq 0.9$ in Lemma 18, $\mathbb{P}[\mathcal{E}_3] \geq 0.9$ in Lemma 19 and $\mathbb{P}[\mathcal{E}_4] \geq 0.97$ in Lemma 20. We also have that $\mathbb{P}[\mathcal{E}_1] \geq 1/3$ due to random agent arrivals and the properties of the binomial distribution. Finally, we argue that $\mathbb{P}[\mathcal{E}_5 | \mathcal{E}_2] = 1/2$. Given that the set of possible realizations is restricted to a subset of those where event \mathcal{E}_2 occurs, \mathcal{E}_5 occurs with probability $1/2$, independently of any other events in the realizations considered in the conditioning on \mathcal{E}_2 . I.e., we can think that once we condition that the event \mathcal{E}_2 occurs, the mechanism flips a fair coin (which is completely independent from \mathcal{E}_2 and also from $\mathcal{E}_1 \cap \mathcal{E}_3 \cap \mathcal{E}_4$): with probability $1/2$, the mechanism selects the right interval (and proceeds collecting value), and with probability $1/2$, the mechanism selects the wrong interval (and does not collect anything). Then,

$$\mathbb{P}[\neg\mathcal{E}_1 \cup \neg\mathcal{E}_2 \cup \neg\mathcal{E}_3 \cup \neg\mathcal{E}_4] \leq 2/3 + 0.1 + 0.1 + 0.03 \leq 0.9$$

Putting everything together: $\mathbb{P}[\mathcal{E}] = \mathbb{P}[\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3 \cap \mathcal{E}_4] \cdot \mathbb{P}[\mathcal{E}_5 | \mathcal{E}_2] \geq \frac{1}{10} \cdot \frac{1}{2} = 1/20$.

4.3 Estimation Testing

The key building block of our LM-Mechanism is an online test, usually referred to as `TestThreshold`, which is applied to a (δ, a, t) -phase. `TestThreshold`, whose pseudocode can be found in Mechanism 2, determines whether:

it is possible to collect value at least $C \cdot a \cdot \hat{t}$, where $C = 1/(7e)$ by applying linear-pricing with threshold \hat{t} to a round with length parameter a .

`TestThreshold` determines the set $\mathcal{N}_{\text{round}}$ of agents of each round by drawing their sizes from a binomial distribution $\mathcal{B}(|\mathcal{N}_{\text{rem}}|, a)$. Then, it offers every agent $b \in \mathcal{N}_{\text{round}}$ the corresponding linear price $p_b = f_S(b)B/\hat{t}$ and collects value $f_S(b)$, if the price p_b is accepted. We write that `TestThreshold` succeeds in a round, if it collects value at least $C \cdot a \cdot \hat{t}$ from the present round, and fails, otherwise. We write that `TestThreshold` succeeds in a phase (or just succeeds) when at least half of the rounds are successful.

`TestThreshold` is the only point where LM-Mechanism makes offers to the agents, collects value and expends budget. Periods 2 and 3 determine the length parameters and the number of phases to which `TestThreshold` is applied, and use their success / fail responses to guide our adaptive search for a good estimate of OPT . Then, Period 4 applies `TestThreshold` to phases with appropriate length parameters in order to collect a total value of $\Omega(\text{OPT})$. If during the execution of LM-Mechanism, either the budget or the agent sequence is exhausted, or an agent with value larger than v_{\max} is found, event \mathcal{E} is violated and `TestThreshold` (and LM-Mechanism) abort (without any value).

We next consider applications of `TestThreshold` to ‘good’ rounds/phases and establish its main properties. We highlight that every time `TestThreshold` is applied in periods 2, 3 and 4, the length parameter a of each round respects the goodness property with respect to the chosen t (see Proposition 1 in Section 6, Proposition 2 in Section 7 and Proposition 3 in Section 8). In fact, this is the key objective behind how the four periods of LM-Mechanism are structured: `PowerTowerSearch`, `BinarySearch` and `Exploitation` are carefully designed in order to ensure that `TestThreshold` is applied to ‘good’ rounds. Therefore, due to Lemma 3 (resp. Lemma 4), every round (resp. phase) to which `TestThreshold` is applied is dense with probability at least 0.9 (resp. close enough to 1).

Algorithm 2: TestThreshold

```
1: Input: Current state  $(S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}})$ , budget  $B$ , phase size  $\delta$ , length parameter  $a$ , threshold  $\hat{t}$ .
2: Initialization:  $C \leftarrow \frac{1}{7e}$ , successes  $\leftarrow 0$ 
3: for  $j = 1$  to  $\delta$  do
4:   Draw  $\tau \sim \mathcal{B}(|\mathcal{N}_{\text{rem}}|, a)$ ,  $\mathcal{N}_{\text{round}} \leftarrow \mathcal{N}_{\text{rem}}[1 : \tau]$ 
5:    $\mathcal{N}_{\text{rem}} \leftarrow \mathcal{N}_{\text{rem}} \setminus \mathcal{N}_{\text{round}}$ ,  $i \leftarrow 1$ , val-test  $\leftarrow 0$ 
6:   if  $\mathcal{N}_{\text{rem}} = \emptyset$  then abort end if
7:   while  $i \leq |\mathcal{N}_{\text{round}}|$  do
8:     if  $f(\mathcal{N}_{\text{round}}[i]) > v_{\max}$  then abort end if
9:     else
10:       $p_i \leftarrow f_S(\mathcal{N}_{\text{round}}[i]) \cdot B / \hat{t}$ 
11:      if  $p_i \geq c_i$  then
12:         $S \leftarrow S \cup \{\mathcal{N}_{\text{round}}[i]\}$ 
13:        val  $\leftarrow \text{val} + f_S(\mathcal{N}_{\text{round}}[i])$  and val-test  $\leftarrow \text{val-test} + f_S(\mathcal{N}_{\text{round}}[i])$ 
14:         $B_{\text{rem}} \leftarrow B_{\text{rem}} - p_i$ 
15:        if  $B_{\text{rem}} \leq 0$  then abort end if
16:      end if
17:    end if
18:    if val-test  $\geq C \cdot a \cdot \hat{t}$  then successes  $\leftarrow$  successes + 1 end if
19:     $i \leftarrow i + 1$ 
20:  end while
21: end for
22: if successes  $\geq \delta/2$  then  $(1, S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}})$  else  $(0, S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}})$ 
```

Due to the linear pricing scheme in step 10 and the success criterion in step 18, every application of TestThreshold to a ‘good’ (a, \hat{t}) -round (regardless of its success or failure) expends a budget of at most

$$\frac{C \cdot a \cdot \hat{t} + v_{\max}}{\hat{t}} \cdot B \leq \left(C \cdot a + \frac{v_{\max}}{\hat{t}} \right) B \leq 2 \cdot C \cdot a \cdot B,$$

where the last inequality holds because the round is good. Thus:

Property 1. When TestThreshold is applied to a ‘good’ (a, \hat{t}) -round the budget expended does not exceed the round’s budget $B_j = 3 \cdot C \cdot a \cdot B$.

Using Property 1, that TestThreshold and that the sum of the length parameters of all rounds is less than 1, we prove the following in Appendix C:

Lemma 6. *The total budget expended by LM-Mechanism is at most $B/10$.*

Property 2. TestThreshold succeeds (i.e., it collects value at least $C \cdot a \cdot \hat{t}$) when applied to an (a_j, \hat{t}_j) -round that is dense and $\hat{t}_j \leq \text{OPT}/4$.

Proof. Let S_j be the part of the solution obtained by applying TestThreshold with threshold \hat{t} to an a_j -round j that is dense with respect to \hat{t} . We assume that the value of each agent b is $f(b) \leq v_{\max}$, as otherwise TestThreshold aborts. Let also S_j^* denote the optimal solution for round j . Suppose some agents $b \in S_j^* \setminus S_j$ were included in the optimal solution but not in S_j . This can happen for one of two reasons: either their private costs exceeded the mechanism’s offer, i.e., $c_b > f_{S_j}(b) B / \hat{t}$, or B_{rem} was insufficient to make an appropriate offer to b . However, by design, only the first case can occur. Due to our linear pricing scheme, no agent is ever offered a price larger than $v_{\max} B / \hat{t} \leq \frac{B}{1024}$, where the inequality holds because we always use thresholds $\hat{t} \geq 1024 v_{\max}$. Combining this with Lemma 6, we conclude that all agents $b \in S_j^* \setminus S_j$ declined our offer.

Algorithm 3: LearningMaxValue

- 1: **Input:** Current set \mathcal{N}_{rem} of agents available
 - 2: Sample $\tau \sim \mathcal{B}(|\mathcal{N}_{\text{rem}}|, 1/3)$, $\mathcal{N}_{\text{period}} \leftarrow \mathcal{N}_{\text{rem}}[1 : \tau]$
 - 3: Offer price $p = 0$ to the agents in $\mathcal{N}_{\text{period}}$
 - 4: Let $v_{\text{max}} \leftarrow \max_{b \in \mathcal{N}_{\text{period}}} f(b)$ and $\mathcal{N}_{\text{rem}} \leftarrow \mathcal{N}_{\text{rem}} \setminus \mathcal{N}_{\text{period}}$
 - 5: **return** $v_{\text{max}}, \mathcal{N}_{\text{rem}}$
-

We can rewrite the sum of the values of these agents as:

$$\sum_{b \in S_j^* \setminus S_j} f_{S_j}(b) = \sum_{b \in S_j^* \setminus S_j} \frac{f_{S_j}(b)}{c_b} \cdot c_b < \sum_{b \in S_j^* \setminus S_j} \frac{\hat{t}}{B} \cdot c_b \leq \hat{t} \cdot \frac{B_j}{B} = 3 \cdot C \cdot a_j \cdot \hat{t} \quad (2)$$

The second to last inequality follows from the hypothesis $c_b > f_{S_j}(b) B / \hat{t}$ and the last inequality follows from the fact that S_j^* is budget-feasible with respect to $B_j = 3 \cdot C \cdot a_j \cdot B$.

Using monotonicity and submodularity of the valuation function we have:

$$f(S_j^*) - f(S_j) \leq f(S_j^* \cup S_j) - f(S_j) \leq \sum_{b \in S_j^* \setminus S_j} f_{S_j}(b) \quad (3)$$

Applying (2) to (3), we get that $f(S_j) \geq f(S_j^*) - 3 \cdot C \cdot a_j \cdot \hat{t}$. Finally, using the hypothesis that round j is dense, and hence $f(S_j^*) \geq C \cdot a_j \cdot \text{OPT}$, we obtain that:

$$f(S_j) \geq C \cdot a_j \cdot \text{OPT} - 3 \cdot C \cdot a_j \cdot \hat{t}$$

Since $\hat{t} \leq \frac{\text{OPT}}{4}$, $f(S_j) \geq 4 \cdot C \cdot a_j \cdot \hat{t} - 3 \cdot C \cdot a_j \cdot \hat{t} = C \cdot a_j \cdot \hat{t}$. Thus, TestThreshold it collects value at least $C \cdot a_j \cdot \hat{t}$ and succeeds.

The Lemma above extends to phases the following way:

Lemma 7. *TestThreshold succeeds (i.e., it collects value at least $C \cdot a \cdot \hat{t}$ and returns 1) when applied to a $(\delta_j, a_j, \hat{t}_j)$ -phase that is dense and $\hat{t}_j \leq \text{OPT}/4$.*

The next four sections discuss how periods 1 to 4 are structured so that TestThreshold is applied to phases so that (conditional on event \mathcal{E}) a total value of $\Omega(\text{OPT})$ is collected.

5 First Period – Learning the Maximum Value

To learn v_{max} , we sample the first τ agents, where $\tau \sim \mathcal{B}(n, 1/3)$. Conditional on event \mathcal{E} , the agent with the maximum value v_{max} is among the first τ agents. In this case, we can use the highest observed value to construct the interval $[v_{\text{max}}, n \cdot v_{\text{max}}]$, which is guaranteed to contain OPT , due to the submodularity of f . The pseudocode for the first period is given in Mechanism 3.

6 Second Period – Power Tower Search

Next, we further refine our estimation of OPT , narrowing it down to two intervals of power tower length. To this end, we partition the interval $[v_{\text{max}}, n \cdot v_{\text{max}}]$ into an interval sequence \mathcal{T} as follows:

$$\mathcal{T} = \{[t_1, t_2], [t_2, t_3], \dots, [t_T, t_{T+1}]\},$$

Algorithm 4: PowerTowerSearch

- 1: **Input:** Current state $(S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}})$, budget B , interval $[v_{\max}, n \cdot v_{\max}]$.
 - 2: **Construct:** sequence \mathcal{T} of interval s.t. $t_1 \leftarrow v_{\max}, t_2 \leftarrow 10^7 \cdot v_{\max}, t_i \leftarrow 2^{\frac{t_i-1}{v_{\max}}} \cdot t_{i-1}$ and $t_{T+1} \leftarrow n \cdot v_{\max}$.
 - 3: **Initialization:** Length parameters $a_i = \frac{81 \cdot e \cdot v_{\max}}{t_i}$ and phase sizes $\gamma_i = \frac{3}{2} \cdot \log\left(\frac{t_i}{v_{\max}}\right)$, index $\leftarrow 1$.
 - 4: **for** $i = 2$ to $|\mathcal{T}|$ **do**
 - 5: $(\text{hit}, S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}) \leftarrow \text{TestThreshold}((S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}), B, \gamma_i, a_i, t_i)$
 - 6: **if** hit = 1 **then** index $\leftarrow i$ **end if**
 - 7: **end for**
 - 8: With probability 1/2: $t_{\min} \leftarrow \mathcal{T}[\text{index} - 1]$ and $t_{\max} \leftarrow \mathcal{T}[\text{index}]$.
 - 9: Otherwise $t_{\min} \leftarrow \mathcal{T}[\text{index}]$ and $t_{\max} \leftarrow \mathcal{T}[\text{index} + 1]$.
 - 10: **return** $t_{\min}, t_{\max}, (S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}})$.
-

where $t_1 = v_{\max}, t_2 = 10^7 v_{\max}, t_i = 2^{\frac{t_i-1}{v_{\max}}} \cdot t_{i-1}$ for $2 < i < T + 1$, and $t_{T+1} = n \cdot v_{\max}$, with $T = |\mathcal{T}|$. Using TestThreshold, we test all intervals $(t_i, t_{i+1}]$, for $i = 2, \dots, T - 1$, so that we find the correct interval to pick. We exclude $[t_1, t_2]$ and $(t_T, t_{T+1}]$, because the correct interval can be inferred without directly testing these values. To test each interval $(t_i, t_{i+1}]$, we apply TestThreshold with threshold $\hat{t} = t_i$ to a (γ_i, a_i, t_i) -phase, where the length parameter is $a_i = \frac{81 \cdot e \cdot v_{\max}}{t_i}$ and the number of rounds is $\gamma_i = \frac{3}{2} \cdot \log\left(\frac{t_i}{v_{\max}}\right)$. The pseudocode of PowerTowerSearch is given in Mechanism 4.

A (a_i, γ_i, t_i) -phase is considered *successful*, if TestThreshold applied with threshold $\hat{t} = t_i$ returns success in at least $\gamma_i/2$ rounds. Each length parameter a_i is chosen such that every (a_i, t_i) -round is ‘good’.

Proposition 1. *During PowerTowerSearch, all phases consist of ‘good’ (a_i, t_i) -rounds.*

The phase lengths γ_i are carefully chosen so that (i) the total number of agents examined during this period is at most a constant fraction of n ; and (ii) we ensure that a phase with threshold much larger than OPT cannot be successful. Formally:

Lemma 8. *Let $\frac{\text{OPT}}{8} \in (t_{i-1}, t_i]$. Then, for every $j \geq i + 1$, the (γ_j, a_j, t_j) -phase tested with threshold $\hat{t} = t_j$ is not successful.*

Proof. Without loss of generality, let $t_i = \frac{\text{OPT}}{8}$. We prove that the $(\gamma_{i+1}, a_{i+1}, t_{i+1})$ -phase tested with threshold $\hat{t} = t_{i+1}$ cannot be successful (if proven for $j = i + 1$, this must also apply to every $j > i + 1$).

If the $(\gamma_{i+1}, a_{i+1}, t_{i+1})$ -phase tested with threshold $\hat{t} = t_{i+1}$ was successful, the total value that we would have collected during that phase would be:

$$\text{Value} \geq \frac{\gamma_{i+1}}{2} \cdot C \cdot a_{i+1} \cdot t_{i+1}$$

Using the definitions of a_i and γ_i , we get that:

$$\text{Value} \geq \frac{\frac{3}{2} \cdot \log\left(\frac{t_{i+1}}{v_{\max}}\right)}{2} \cdot C \cdot \frac{81 \cdot e \cdot v_{\max}}{t_{i+1}} \cdot t_{i+1}$$

Since $t_{i+1} = 2^{t_i/v_{\max}} \cdot t_i$ and using that $C = 1/(7e)$, we obtain that:

$$\text{Value} > 8 \cdot \log\left(2^{\frac{t_i}{v_{\max}}} \cdot \frac{t_i}{v_{\max}}\right) \cdot v_{\max}$$

Since $2^{\frac{t_i}{v_{\max}}} \cdot \frac{t_i}{v_{\max}} \geq 2^{\frac{t_i}{v_{\max}}}$, we get $\text{Value} > 8 \cdot t_i = \text{OPT}$, a contradiction to the definition of OPT.

After testing all (γ_i, a_i, t_i) -phases corresponding to intervals $(t_i, t_{i+1}]$, each with threshold t_i , we select the interval corresponding to the last successful phase and the interval preceding it. I.e., if $(t_i, t_{i+1}]$ corresponds to the last successful phase, we select the intervals $(t_{i-1}, t_i]$ and $(t_i, t_{i+1}]$. In the definition of event \mathcal{E}_2 , we require that either $\frac{\text{OPT}}{8} \in (t_{i-1}, t_i]$ or $\frac{\text{OPT}}{8} \in (t_i, t_{i+1}]$, where $(t_{i-1}, t_i]$ and $(t_i, t_{i+1}]$ are the intervals selected by PowerTowerSearch. Let i^* be such that $\frac{\text{OPT}}{8} \in (t_{i^*}, t_{i^*+1}]$. Then, for the event \mathcal{E}_2 to occur, we require that either the $(\gamma_{i^*}, a_{i^*}, t_{i^*})$ -phase or the $(\gamma_{i^*+1}, a_{i^*+1}, t_{i^*+1})$ -phase is the last successful phase.

We next show that the event \mathcal{E}_2 is implied by the event that the $(\gamma_{i^*}, a_{i^*}, t_{i^*})$ -phase is dense. So, let us assume that the $(\gamma_{i^*}, a_{i^*}, t_{i^*})$ -phase is dense, which occurs with probability at least 0.9, by Lemma 18 (which, in turn, follows from Lemma 4). Hence, with probability at least 0.9, the $(\gamma_{i^*}, a_{i^*}, t_{i^*})$ -phase is successful. If the $(\gamma_{i^*}, a_{i^*}, t_{i^*})$ -phase is the last successful phase, then we select the intervals $(t_{i^*-1}, t_{i^*}]$ and $(t_{i^*}, t_{i^*+1}]$ and the event \mathcal{E}_2 occurs. Otherwise, Lemma 8 implies that the $(\gamma_{i^*+1}, a_{i^*+1}, t_{i^*+1})$ -phase must be the last successful phase. Then, we select the intervals $(t_{i^*}, t_{i^*+1}]$ and $(t_{i^*+1}, t_{i^*+2}]$ and the event \mathcal{E}_2 occurs again.

7 Third Period – Binary Search

We next show how to apply *binary search* to the interval returned by Period 2. To apply binary search to the given interval $(t_{\min}, t_{\max}]$, it is essential that the feedback on each estimation is correct with high probability. To achieve this, we test each estimation \hat{t} of the optimum on (m, a, \hat{t}) -phases, where $m = \left\lceil 8 \cdot \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil$ rounds and $a = \frac{1}{6 \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil \cdot m}$. The value of m is set so that we can lower bound the probability of the event \mathcal{E}_3 by union bound on the number of different phases. The length parameter a is set so that the total number of phases used in Period 3 times the number m of rounds in each phase times a is at most $1/3$ (which implies that Period 3 “consumes” about $1/3$ of the agent sequence). We next show that this choice of a makes our rounds good:

Proposition 2. *(a, t_{\min}) -rounds (and thus all rounds) during BinarySearch are good.*

Proof. We need to verify that $\frac{t_{\min}}{v_{\max}} \geq \frac{81e}{a}$. Using our choice of a , we obtain that:

$$\frac{t_{\min}}{v_{\max}} \geq 486 \cdot e \cdot \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil \cdot \left\lceil 8 \cdot \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil$$

Using the fact that $t_{\max} = 2 \frac{t_{\min}}{v_{\max}} \cdot t_{\min}$, we get:

$$\frac{t_{\min}}{v_{\max}} \geq 486 \cdot e \cdot \left\lceil \log \left(\frac{t_{\min}}{v_{\max}} \right) \right\rceil \cdot \left\lceil 8 \cdot \log \left(\frac{t_{\min}}{v_{\max}} \right) \right\rceil,$$

which is true for $\frac{t_{\min}}{v_{\max}} \geq 10^7$.

The number of rounds needed to conduct BinarySearch is $\ell = \frac{1}{6 \cdot a}$. The pseudocode for BinarySearch is presented in Mechanism 5.

By Lemma 7, if all thresholds not greater than $\frac{\text{OPT}}{4}$ are tested on dense phases, we cannot end up with a substantial underestimation of OPT after conducting BinarySearch. Below we prove an even sharper bound on the possible estimates that BinarySearch may return.

Lemma 9. *Let \hat{t} be the estimate BinarySearch returns, under event \mathcal{E} . Then:*

$$\frac{\text{OPT}}{8} \leq \hat{t} \leq 84 \cdot e \cdot \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil \cdot \text{OPT}$$

Algorithm 5: BinarySearch

1: **Input:** Current state $(S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}})$, budget B , search interval $[t_{\min}, t_{\max}]$
2: **Initialization:** Phase size $m \leftarrow \left\lceil 8 \cdot \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil$, length parameter $a \leftarrow \frac{1}{6 \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil \cdot m}$,
 $\text{low} \leftarrow 0$, $\text{high} \leftarrow \left\lceil \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil$, $\text{mid} \leftarrow \lceil (\text{high} + \text{low})/2 \rceil$
3: **while** $\text{low} \leq \text{high}$ **do**
4: $\text{hit} \leftarrow 0$
5: $(\text{hit}, S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}) \leftarrow \text{TestThreshold}((S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}), B, m, a, 2^{\text{mid}} \cdot t_{\min})$
6: **if** $\text{hit} = 1$ **then** $\text{low} \leftarrow \text{mid}$, $\text{mid} \leftarrow \lceil (\text{high} + \text{low})/2 \rceil$.
7: **else** $\text{high} \leftarrow \text{mid}$, $\text{mid} \leftarrow \lfloor (\text{high} + \text{low})/2 \rfloor$ **end if**
8: **end while**
9: **return** $(2^{\text{mid}} \cdot t_{\min}, (S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}))$.

Algorithm 6: Exploitation

1: **Input:** Current state $(S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}})$, budget B , initial threshold t_{init} , search interval $[t_{\min}, t_{\max}]$
2: **Initialization:** $\hat{t} \leftarrow t_{\text{init}}$, Phase size $m \leftarrow \left\lceil 8 \cdot \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil$, length parameter $a \leftarrow \frac{1}{6 \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil \cdot m}$
3: **for** $i = 1$ to $\frac{1}{6 \cdot m \cdot a} = \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil$ **do**
4: $(\text{hit}, S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}) \leftarrow \text{TestThreshold}((S, \text{val}, \mathcal{N}_{\text{rem}}, B_{\text{rem}}), B, a, \hat{t})$
5: **if** $\text{hit} = 1$ **then** $\hat{t} \leftarrow 2 \cdot \hat{t}$ **else** $\hat{t} \leftarrow \hat{t}/2$ **end if**
6: **end for**
7: **return** (S, val) .

Proof. Suppose $\hat{t} < \frac{\text{OPT}}{8}$. This would imply that an intermediate estimate $\hat{t}' \in \left[\frac{\text{OPT}}{8}, \frac{\text{OPT}}{4} \right]$ failed in a dense phase. However, this cannot happen under event \mathcal{E} , due to Lemma 7. Now consider the case where $\hat{t} > 84e \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil \cdot \text{OPT}$. This would mean that a dense phase succeeded using an overestimated value $\hat{t}' > 84e \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil \cdot \text{OPT}$. Specifically, in that phase, the accumulated value would be:

$$\text{Value} \geq \frac{m}{2} \cdot C \cdot a \cdot \hat{t}'$$

By using the assumption of the overestimation \hat{t}' , along with the definitions of C and a we get:

$$\text{Value} > \frac{m}{2} \cdot \frac{1}{7e} \cdot \frac{1}{6 \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil \cdot m} \cdot 84e \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil \cdot \text{OPT} = \text{OPT}$$

which clearly contradicts the definition of OPT.

Thus, we conclude that $\hat{t} \in \left[\frac{\text{OPT}}{8}, 84e \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \cdot \text{OPT} \right]$

8 Fourth Period – Exploitation

Lemma 9 indicates that, after running BinarySearch, our estimation is at most $84 \cdot e \cdot \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}} \right) \right\rceil$ times OPT. In the final stage of our mechanism, we aim to converge to a threshold \hat{t} close to OPT and apply \hat{t} to a constant fraction of agents, thereby collecting a total value of $\Omega(\text{OPT})$.

Exploitation, whose pseudocode is presented in Mechanism 6, is applied across a total number of ℓ/m (m, a, \hat{t}) -phases, thus using $\ell = 1/(6a)$ rounds in total. Under the event \mathcal{E} , it holds that $t_{\min} \leq \frac{\text{OPT}}{8}$. Thus, we can leverage Proposition 2 to claim the following:

Proposition 3. *Conditioning on the event \mathcal{E} , all rounds during Exploitation are good.*

The mechanism operates by doubling the estimation after every successful phase and halving it after every failed phase. Under the event \mathcal{E} , the threshold \hat{t} never falls below $\text{OPT}/8$, by Lemma 7. Combining this with Lemma 9, we conclude that we can have no more than

$$\frac{\frac{\ell}{m} + \log \log \log \left(\frac{t_{\max}}{t_{\min}} \right) + \log(84e) + 2}{2}$$

failed phases; exceeding this would cause our estimation to drop below $\frac{\text{OPT}}{8}$. Consequently, we are guaranteed at least $\frac{\frac{\ell}{m} - \log \log \log \left(\frac{t_{\max}}{t_{\min}} \right) - \log(84e) - 2}{2}$ successful phases, which ensures a constant competitive ratio. Formally,

Lemma 10. *Conditional on the event \mathcal{E} , Exploitation collects a total value of at least $\frac{\text{OPT}}{4032e}$.*

Proof. As we have argued already, we are guaranteed to have at least $\frac{\frac{\ell}{m} - \log \log \log \left(\frac{t_{\max}}{t_{\min}} \right) - \log(84e) - 2}{2}$ successful Phases, which in turn means that the value gathered is greater than:

$$\text{Value} \geq \underbrace{\frac{\frac{\ell}{m} - \log \log \log \left(\frac{t_{\max}}{t_{\min}} \right) - \log(84e) - 2}{2}}_{\text{successful Phases}} \cdot \underbrace{\frac{m}{2} \cdot C \cdot a \cdot \frac{\text{OPT}}{8}}_{\text{Value of each Phase}}$$

Now using the definitions of a , ℓ , C , and t_{\max} , we get that

$$\text{Value} \geq \left(\frac{1}{6} - \frac{\log \log \log \left(2 \frac{t_{\min}}{v_{\max}} \right) + \log(84e) + 2}{6 \cdot \log \log \left(2 \frac{t_{\min}}{v_{\max}} \right)} \right) \cdot \frac{\text{OPT}}{224e}$$

Finally, by the monotonicity of $g(x) = \frac{-\log \log \log(x) - \log(84e) - 2}{6 \cdot \log \log(x)}$, along with the fact that $\frac{t_{\min}}{v_{\max}} \geq 10^7$, we conclude that $\text{Value} \geq \frac{1}{4032e} \cdot \text{OPT}$.

9 Putting Everything Together and Removing the Large Market Assumption

Below we present the proof of the competitiveness of LM-Mechanism.

Theorem 3. *Assuming that $\text{OPT} > 10^7 \cdot v_{\max}$, LM-Mechanism is $O(1)$ -competitive.*

Proof. In the four periods of LM-Mechanism, we assume that the event \mathcal{E} occurs. Lemma 5 proves that \mathcal{E} happens with probability at least $1/20$. Conditional on the event \mathcal{E} , the LM-Mechanism achieves a competitive ratio of $4032e$ due to Lemma 10. Overall, the expected total value of LM-Mechanism is at least $\frac{1}{20} \cdot \frac{1}{4032e} \cdot \text{OPT}$.

Finally, we need to remove the Large Market Assumption that $\text{OPT} > 10^7 \cdot v_{\max}$. To this end, we present the mechanism PostedPrices, whose pseudocode can be found in Mechanism 7. PostedPrices invokes three different mechanisms, with a constant probability each, which deal with instances of different market size (i.e., magnitude of $\frac{\text{OPT}}{v_{\max}}$). We are now ready to prove our main result:

Theorem 4. *PostedPrices is a universally truthful $O(1)$ -competitive posted-price mechanism for online budget-feasible procurement auctions with secretary agent arrivals and monotone submodular buyer's valuations.*

Algorithm 7: PostedPrices

- 1: **Input:** Set of agents \mathcal{N} , budget B
 - 2: With probability 0.1, execute Dynkin’s algorithm on \mathcal{N} .
 - 3: With probability 0.1, execute MediumMarket on the set of agents \mathcal{N} with budget B .
 - 4: With probability 0.8, execute LM-Mechanism on the set of agents \mathcal{N} with budget B .
-

Algorithm 8: MediumMarket

- 1: **Input:** Set of agents \mathcal{N} , budget B
 - 2: Learn v_{\max} using LearningMaxValue
 - 3: Pick uniformly at random a threshold $t \in \{2^6 \cdot v_{\max}, 2^8 \cdot v_{\max}, \dots, 2^{23} \cdot v_{\max}\}$
 - 4: Apply linear pricing with threshold t and budget B to the rest of the agent sequence.
-

Proof. Universal truthfulness follows from the fact that PostedPrices is a probability distribution over three universally truthful (and posted-price) mechanisms.

Regarding the competitive ratio of PostedPrices, we consider the following cases:

1. $\text{OPT} < 1024 \cdot v_{\max}$: Then Dynkin’s algorithm is executed with probability 0.1 and collects an expected value of at least $\frac{1}{10} \cdot \frac{\text{OPT}}{1024 \cdot e}$.
2. $1024 \cdot v_{\max} \leq \text{OPT} < 10^7 \cdot v_{\max}$: Then MedianMarket is executed with probability 0.1. With probability at least 0.9, the optimal value for the set of agents not used for the calculation of v_{\max} by LearningMaxValue, in step 2, is at least $\text{OPT}/4$. With probability at least $1/18$, the threshold t selected is such that $\frac{\text{OPT}}{16} \leq t \leq \frac{\text{OPT}}{8}$. Thus, by Lemma 12, the chosen threshold t results in at least $(\frac{1}{4} - \frac{v_{\max}}{\text{OPT}}) \cdot \frac{\text{OPT}}{4} > \frac{\text{OPT}}{18}$ value. Overall the expected total value is at least:

$$\frac{1}{10} \cdot \frac{9}{10} \cdot \frac{1}{18} \cdot \frac{\text{OPT}}{18} = \frac{\text{OPT}}{3600}$$

3. $10^7 \cdot v_{\max} \leq \text{OPT}$: Then, LM-Mechanism is executed with probability 0.8, which by Theorem 3 results in an expected total value of at least $0.8 \cdot \frac{1}{20} \cdot \frac{\text{OPT}}{4032e}$.

Putting the three cases together, we conclude that the competitive ratio of PostedPrices is $O(1)$. We note that at many places, we prioritized simplicity over trying to optimize the resulting constant.

10 Conclusions

In this work, we have introduced a randomized, constant-competitive posted-price mechanism for online procurement auctions, thus resolving the main open question of Badanidiyuru, Kleinberg and Singer [17]. Our findings demonstrate that in online procurement auctions, sequential posted-price mechanisms can be as powerful, in terms of the asymptotics of their competitive ratio, as seal-bid mechanisms, whose performance has been extensively studied. We note that despite our mechanism is elaborate to design and analyze, its interface with the agents is simple and transparent.

In addition to improving the constants in our analysis, an interesting direction for further research is whether our refined adaptive search can be generalized and applied to other posted-price mechanisms, such as combinatorial auctions with submodular bidders [35,36,37], towards improved competitive ratios.

References

1. Y. Singer, “Budget feasible mechanisms,” in *Proc. of the 51th IEEE Symposium on Foundations of Computer Science (FOCS 2010)*. IEEE Computer Society, 2010, pp. 765–774.

2. G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions - I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.
3. S. Khuller, A. Moss, and J. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.
4. M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Operations Research Letters*, vol. 32, no. 1, pp. 41–43, 2004.
5. A. Badanidiyuru, S. Dobzinski, and S. Oren, "Optimization with demand oracles," *Algorithmica*, vol. 81, no. 6, pp. 2244–2269, 2019.
6. Y. Singer, "Budget feasible mechanism design," *SIGecom Exchanges*, vol. 12, no. 2, pp. 24–31, 2013.
7. Y. Singer and M. Mittal, "Pricing mechanisms for crowdsourcing markets," in *Proc. of the 22nd International World Wide Web Conference (WWW 2013)*. ACM, 2013, pp. 1157–1166.
8. A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," in *Proc. of the 22nd International World Wide Web Conference (WWW 2013)*. ACM, 2013, pp. 1167–1178.
9. Q. Liu, T. Luo, R. Tang, and S. Bressan, "An efficient and truthful pricing mechanism for team formation in crowdsourcing markets," in *Proc. of the IEEE Conference on Communications (ICC 2015)*, 2015, pp. 567–572.
10. Z. Zheng, S. Yang, J. Xie, F. Wu, X. Gao, and G. Chen, "On Designing Strategy-Proof Budget Feasible Online Mechanisms for Mobile Crowdsensing With Time-Discounting Values," *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 2088–2102, 2022.
11. F. Restuccia, S. K. Das, and J. Payton, "Incentive Mechanisms for Participatory Sensing: Survey and Research Challenges," *ACM Transactions on Sensor Networks*, vol. 12, no. 2, 2016.
12. X. Liu, H. Chan, M. Li, and W. Wu, "Budget feasible mechanisms: A survey," in *Proc. of the 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*. ijcai.org, 2024, pp. 8132–8141.
13. N. Gravin, Y. Jin, P. Lu, and C. Zhang, "Optimal budget-feasible mechanisms for additive valuations," *ACM Transactions on Economics and Computation*, vol. 8, no. 4, pp. 21:1–21:15, 2020.
14. P. Jalaly and É. Tardos, "Simple and efficient budget feasible mechanisms for monotone submodular valuations," *ACM Transactions on Economics and Computation*, vol. 9, no. 1, pp. 4:1–4:20, 2021.
15. E. Balkanski, P. Garimidi, V. Gkatzelis, D. Schoepflin, and X. Tan, "Deterministic budget-feasible clock auctions," in *Proc. of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA 2022)*. SIAM, 2022, pp. 2940–2963.
16. G. Amanatidis, P. Kleer, and G. Schäfer, "Budget-feasible mechanism design for non-monotone submodular objectives: Offline and online," *Mathematics of Operations Research*, vol. 47, no. 3, pp. 2286–2309, 2022.
17. A. Badanidiyuru, R. Kleinberg, and Y. Singer, "Learning on a budget: posted price mechanisms for online procurement," in *Proc. of the 13th ACM Conference on Electronic Commerce (EC 2012)*. ACM, 2012, pp. 128–145.
18. S. Li, "Obviously strategy-proof mechanisms," *The American Economic Review*, vol. 107, no. 1, pp. 3257–3287, 2017.
19. E. Balkanski and J. D. Hartline, "Bayesian budget feasibility with posted pricing," in *Proc. of the 25th Conference on World Wide Web (WWW 2016)*. ACM, 2016, pp. 189–203.
20. S. Chawla, J. D. Hartline, D. L. Malec, and B. Sivan, "Multi-parameter mechanism design and sequential posted pricing," in *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC 2010)*. ACM, 2010, pp. 311–320.
21. E. B. Dynkin, "The optimum choice of the instant for stopping a Markov process," *Soviet Mathematics - Doklady*, vol. 4, pp. 627–629, 1963.
22. N. Chen, N. Gravin, and P. Lu, "On the approximability of budget feasible mechanisms," in *Proc. of the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA 2011)*. SIAM, 2011, pp. 685–699.
23. N. Anari, G. Goel, and A. Nikzad, "Mechanism design for crowdsourcing: An optimal $1 - 1/e$ competitive budget-feasible mechanism for large markets," in *Proc. of the 55th IEEE Symposium on Foundations of Computer Science (FOCS 2014)*. IEEE Computer Society, 2014, pp. 266–275.
24. K. Han, Y. Wu, H. Huang, and S. Cui, "Triple eagle: Simple, fast and practical budget-feasible mechanisms," in *Proc. of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023.
25. H. Huang, K. Han, S. Cui, and J. Tang, "Randomized pricing with deferred acceptance for revenue maximization with submodular objectives," in *Proc. of the 32nd ACM Web Conference 2023, (WWW 2023)*. ACM, 2023, pp. 3530–3540.
26. X. Bei, N. Chen, N. Gravin, and P. Lu, "Budget feasible mechanism design: from prior-free to bayesian," in *Proc. of the 44th Symposium on Theory of Computing Conference (STOC 2012)*. ACM, 2012, pp. 449–458.
27. G. Amanatidis, G. Birmpas, and E. Markakis, "On budget-feasible mechanism design for symmetric submodular objectives," in *Proc. of the 13th Conference on Web and Internet Economics (WINE 2017)*, ser. Lecture Notes in Computer Science, vol. 10660. Springer, 2017, pp. 1–15.
28. S. Dobzinski, C. H. Papadimitriou, and Y. Singer, "Mechanisms for complement-free procurement," in *Proc. of the 12th ACM Conference on Electronic Commerce (EC 2011)*. ACM, 2011, pp. 273–282.
29. M. Bateni, M. T. Hajiaghayi, and M. Zadimoghaddam, "Submodular secretary problem and extensions," *ACM Transactions on Algorithms*, vol. 9, no. 4, pp. 32:1–32:23, 2013.

30. T. Kesselheim and A. Tönnis, “Submodular secretary problems: Cardinality, matching, and linear constraints,” in *Proc. of the 20th Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2017) and of the 21st Conference on Randomization and Computation (RANDOM 2017)*, ser. LIPIcs, vol. 81. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 16:1–16:22.
31. M. Feldman, J. Naor, and R. Schwartz, “Improved competitive ratios for submodular secretary problems (extended abstract),” in *Proc. of the 14th Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2011) and of the 15th Conference on Randomization and Computation (RANDOM 2011)*, ser. Lecture Notes in Computer Science, vol. 6845. Springer, 2011, pp. 218–229.
32. A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*. Cambridge University Press, 1998.
33. A. C. Yao, “Probabilistic Computations: Toward a Unified Measure of Complexity (Extended Abstract),” in *Proc. of the 18th Symposium on Foundations of Computer Science (FOCS 1977)*. IEEE Computer Society, 1977, pp. 222–227.
34. R. D. Kleinberg, “A multiple-choice secretary algorithm with applications to online auctions,” in *Proc. of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*. SIAM, 2005, pp. 630–631.
35. S. Assadi and S. Singla, “Improved truthful mechanisms for combinatorial auctions with submodular bidders,” in *Proc. of the 60th IEEE Symposium on Foundations of Computer Science (FOCS 2019)*. IEEE Computer Society, 2019, pp. 233–248.
36. ———, “Improved truthful mechanisms for combinatorial auctions with submodular bidders,” *SIGecom Exchanges*, vol. 18, no. 1, pp. 19–27, 2020.
37. P. Dütting, T. Kesselheim, and B. Lucier, “An $o(\log \log m)$ prophet inequality for subadditive combinatorial auctions,” *SIAM Journal on Computing*, vol. 53, no. 6, pp. 239–275, 2024.
38. N. L. Johnson and S. Kotz., *Urn Models and Their Application: An Approach to Modern Discrete Probability Theory*, ser. Approach to Modern Discrete Probability Theory. Wiley, 1977.
39. D. Dubhashi and D. Ranjan, “Balls and bins: a study in negative dependence,” *Random Structures and Algorithms*, vol. 13, no. 2, p. 99–124, Sep. 1998.

Appendix

A Detailed Proof of Theorem 2

We consider a family of instances $\mathcal{I} = \{I_0, I_1, \dots, I_{\log n}\}$. Every instance consists of n value-private cost pairs (v, c) , where for instance I_i , $v = 1$ and $c = B/2^i$, for every $i \in \{0, 1, \dots, \log n\}$ (i.e., I_i consists of n value-cost pairs $(1, B/2^i)$).

We construct a probability distribution $\mathcal{D}_{\mathcal{I}}$ over the family \mathcal{I} of instances such that the expected value of any deterministic non-adaptive linear-price mechanism on $\mathcal{D}_{\mathcal{I}}$ is at most 1, while the expected value of the optimal solution on $\mathcal{D}_{\mathcal{I}}$ is $\frac{\log n}{2} + 1$. Then, we apply Yao's principle, which extends the lower bound to any randomized mechanism ([32, Chapter 8.4] and [33]).

Specifically, for every $i = 0, 1, 2, \dots, \log(n) - 1$, we let I_i appear with probability $p_i = 1/2^{i+1}$ in $\mathcal{D}_{\mathcal{I}}$. We let $I_{\log n}$ appear with probability $p_{\log n} = 1/n$ in $\mathcal{D}_{\mathcal{I}}$, so that the sum of p_i 's is equal to 1. The optimal solution in each I_i is 2^i . Therefore, the expected value of the optimal solution is

$$\mathbb{E}[\text{OPT}] = \frac{n}{n} + \sum_{i=0}^{\log(n)-1} \frac{2^i}{2^{i+1}} = \frac{\log n}{2} + 1.$$

We let now fix any deterministic non-adaptive linear-price mechanism ALG. For simplicity and without loss of generality, we assume that ALG is aware of v in advance (clearly, this can only work in ALG's favor). Since ALG knows v and does not get any information about the agents' private costs before it makes its first offer, ALG can be regarded as selecting an arbitrary fixed index $i \in \{0, 1, \dots, \log n\}$ and an arbitrary fixed threshold $\hat{t} \in [2^i, 2^{i+1})$. Then, ALG determines its linear-price, which is the same for every agent.

For simplicity, we let ALG's fixed price be $B/2^i$, for some fixed $i \in \{0, 1, \dots, \log n\}$. Then, ALG's value for instance I_j is 0, if $j < i$, and 2^i for all $j = i, \dots, \log n$. Therefore, the expected value of ALG's solution is

$$\mathbb{E}[\text{ALG}] = \frac{2^i}{n} + 2^i \sum_{j=i}^{\log(n)-1} \frac{1}{2^{j+1}} = \frac{2^i}{n} + 2^i \left(\frac{1}{2^i} - \frac{1}{n} \right) = 1.$$

Then, the theorem follows from Yao's principle. □

B Proofs Missing from Section 4.1

B.1 Partitioning into Rounds

Suppose we wish to construct κ rounds with length parameters $\{a_1, a_2, \dots, a_\kappa\}$. We will analytically compute the participation probability of the κ -th round.

Proof (Proof of Lemma 1:). Consider an agent b and let R_κ denote the set of agents in the κ -th round, then:

$$\mathbb{P}[b \in R_\kappa] = \sum_{x_1=0}^{n_1} \mathbb{P}[x_1] \sum_{x_2=0}^{n_2} \dots \sum_{x_{\kappa-1}=0}^{n_{\kappa-1}} \mathbb{P}[x_{\kappa-1}] \sum_{x_\kappa=1}^{n_\kappa} \mathbb{P}[x_\kappa] \cdot \mathbb{P}[b \in R_\kappa | x_1, \dots, x_\kappa].$$

Where $n_1 = n$ and $n_i = n - \sum_{j=1}^{i-1} x_j$. By using the fact that the order of the agents follows a uniform distribution we get that:

$$\mathbb{P}[b \in R_\kappa] = \sum_{x_1=0}^{n_1} \mathbb{P}[x_1] \sum_{x_2=0}^{n_2} \dots \sum_{x_{\kappa-1}=0}^{n_{\kappa-1}} \mathbb{P}[x_{\kappa-1}] \sum_{x_\kappa=1}^{n_\kappa} \mathbb{P}[x_\kappa] \cdot \frac{x_\kappa}{n}.$$

By definition, $\sum_{x_\kappa=1}^{n_\kappa} \mathbb{P}[x_\kappa] \cdot x_\kappa = \mathbb{E}[\kappa\text{-th round length}] = a_\kappa \cdot n_\kappa$, thus:

$$\mathbb{P}[b \in R_\kappa] = \sum_{x_1=0}^{n_1} \mathbb{P}[x_1] \sum_{x_2=0}^{n_2} \cdots \sum_{x_{\kappa-1}=0}^{n_{\kappa-1}} \mathbb{P}[x_{\kappa-1}] \frac{n_\kappa}{n} \cdot a_\kappa.$$

Observing that $n_\kappa = n_{\kappa-1} - x_{\kappa-1}$ and applying the same argument as before repeatedly, we finally get:

$$\mathbb{P}[b \in R_\kappa] = (1 - a_1) \cdot (1 - a_2) \cdots (1 - a_{\kappa-1}) \cdot a_\kappa.$$

Now it remains to prove the independency among agents.

Suppose 2 agents b_1, b_2 , then:

$$\mathbb{P}[\{b_1, b_2\} \in R_\kappa] = \sum_{x_1=0}^{n_1-2} \mathbb{P}[x_1] \sum_{x_2=0}^{n_2-2} \cdots \sum_{x_{\kappa-1}=0}^{n_{\kappa-1}-2} \mathbb{P}[x_{\kappa-1}] \sum_{x_\kappa=2}^{n_\kappa} \mathbb{P}[x_\kappa] \cdot \mathbb{P}[\{b_1, b_2\} \in R_\kappa | x_1, \dots, x_\kappa].$$

Since agent order follows a uniform distribution and the round lengths are drawn from a binomial distribution, we get:

$$\mathbb{P}[\{b_1, b_2\} \in R_\kappa] = \sum_{x_1=0}^{n_1-2} \mathbb{P}[x_1] \sum_{x_2=0}^{n_2-2} \cdots \sum_{x_{\kappa-1}=0}^{n_{\kappa-1}-2} \mathbb{P}[x_{\kappa-1}] \sum_{x_\kappa=2}^{n_\kappa} \binom{n_\kappa}{x_\kappa} \cdot a_\kappa^{x_\kappa} \cdot (1 - a_\kappa)^{n_\kappa - x_\kappa} \cdot \frac{\binom{x_\kappa}{2}}{\binom{n}{2}}.$$

Using the identity:

$$\binom{n_\kappa}{x_\kappa} \cdot \binom{x_\kappa}{2} = \binom{n_\kappa}{2} \cdot \binom{n_\kappa - 2}{x_\kappa - 2},$$

we obtain:

$$\mathbb{P}[\{b_1, b_2\} \in R_\kappa] = \sum_{x_1=0}^{n_1-2} \mathbb{P}[x_1] \sum_{x_2=0}^{n_2-2} \cdots \sum_{x_{\kappa-1}=0}^{n_{\kappa-1}-2} \mathbb{P}[x_{\kappa-1}] \cdot \frac{\binom{n_\kappa}{2}}{\binom{n}{2}} \sum_{x_\kappa=2}^{n_\kappa} \binom{n_\kappa - 2}{x_\kappa - 2} \cdot a_\kappa^{x_\kappa} \cdot (1 - a_\kappa)^{n_\kappa - x_\kappa}.$$

Noting that:

$$\sum_{x_\kappa=2}^{n_\kappa} \binom{n_\kappa - 2}{x_\kappa - 2} \cdot a_\kappa^{x_\kappa} \cdot (1 - a_\kappa)^{n_\kappa - x_\kappa} = a_\kappa^2 \cdot \sum_{x_\kappa=0}^{n_\kappa-2} \binom{n_\kappa - 2}{x_\kappa} \cdot a_\kappa^{x_\kappa} \cdot (1 - a_\kappa)^{n_\kappa - 2 - x_\kappa} = a_\kappa^2.$$

and thus:

$$\mathbb{P}[\{b_1, b_2\} \in R_\kappa] = \sum_{x_1=0}^{n_1-2} \mathbb{P}[x_1] \sum_{x_2=0}^{n_2-2} \cdots \sum_{x_{\kappa-1}=0}^{n_{\kappa-1}-2} \mathbb{P}[x_{\kappa-1}] \cdot \frac{\binom{n_\kappa}{2}}{\binom{n}{2}} \cdot a_\kappa^2.$$

Now we focus on the last sum,

$$\sum_{x_{\kappa-1}=0}^{n_{\kappa-1}-2} \mathbb{P}[x_{\kappa-1}] \cdot \frac{\binom{n_\kappa}{2}}{\binom{n}{2}} \cdot a_\kappa^2 = a_\kappa^2 \sum_{x_{\kappa-1}=0}^{n_{\kappa-1}-2} \binom{n_{\kappa-1}}{x_{\kappa-1}} \cdot a_{\kappa-1}^{x_{\kappa-1}} \cdot (1 - a_{\kappa-1})^{n_{\kappa-1} - x_{\kappa-1}} \cdot \frac{\binom{n_\kappa}{2}}{\binom{n}{2}}. \quad (4)$$

We notice that:

$$\binom{n_{\kappa-1}}{x_{\kappa}} \cdot \frac{\binom{n_{\kappa}}{2}}{\binom{n}{2}} = \binom{n_{\kappa-1}-2}{x_{\kappa-1}} \cdot \frac{\binom{n_{\kappa-1}}{2}}{\binom{n}{2}}, \quad (5)$$

and by applying (5) to (4) we get:

$$\sum_{x_{\kappa-1}=0}^{n_{\kappa-1}-2} \mathbb{P}[x_{\kappa-1}] \cdot \frac{\binom{n_{\kappa}}{2}}{\binom{n}{2}} \cdot a_{\kappa}^2 = a_{\kappa}^2 \frac{\binom{n_{\kappa-1}}{2}}{\binom{n}{2}} \cdot \sum_{x_{\kappa-1}=0}^{n_{\kappa-1}-2} \binom{n_{\kappa-1}-2}{x_{\kappa-1}} \cdot a_{\kappa-1}^{x_{\kappa-1}} \cdot (1-a_{\kappa-1})^{n_{\kappa-1}-x_{\kappa-1}}. \quad (6)$$

Next we notice that:

$$\sum_{x_{\kappa-1}=0}^{n_{\kappa-1}-2} \binom{n_{\kappa-1}-2}{x_{\kappa-1}} \cdot a_{\kappa-1}^{x_{\kappa-1}} \cdot (1-a_{\kappa-1})^{n_{\kappa-1}-x_{\kappa-1}} = (1-a_{\kappa-1})^2, \quad (7)$$

and combining (6), (7) we get that:

$$\sum_{x_{\kappa-1}=0}^{n_{\kappa-1}-2} \mathbb{P}[x_{\kappa-1}] \cdot \frac{\binom{n_{\kappa}}{2}}{\binom{n}{2}} \cdot a_{\kappa}^2 = \frac{\binom{n_{\kappa-1}}{2}}{\binom{n}{2}} \cdot (1-a_{\kappa-1})^2 \cdot a_{\kappa}^2$$

It is now evident that by following the same procedure for the rest of the sums we get that:

$$\mathbb{P}[\{b_1, b_2\} \in R_{\kappa}] = a_{\kappa}^2 \cdot \prod_{i=1}^{\kappa-1} (1-a_i)^2 = \mathbb{P}[b_1 \in R_{\kappa}] \cdot \mathbb{P}[b_2 \in R_{\kappa}]$$

This confirms that agent selections are independent.

Lemma 1 also proves that rounds are indeed equivalent with a random set of the whole input where each agent is included with the same probability.

An important property of the lengths drawn in Period 2 that will turn out very useful in the rest of our analysis is the following:

Lemma 11. *For every $i \in \{2, \dots, T\}$ (where T is the number of intervals in the partitioning of $[v_{\max}, n \cdot v_{\max}]$ used in Section 6), it holds that:*

$$a_i \cdot \gamma_i \leq \frac{1}{i^{10}}$$

Proof. We first notice the following:

$$a_i \cdot \gamma_i = 81e \cdot \frac{v_{\max}}{t_i} \cdot \frac{3}{2} \cdot \log\left(\frac{t_i}{v_{\max}}\right)$$

Applying the fact that $\log\left(\frac{t_i}{v_{\max}}\right) < \frac{t_i-1}{v_{\max}}$, we get:

$$a_i \cdot \gamma_i < 122e \cdot \frac{t_i-1}{t_i}$$

Now consider the following sequences of real numbers:

$$h_{1,i} = 122e \cdot \frac{t_{i-1}}{t_i}$$

$$h_{2,i} = \frac{1}{i^{10}}$$

we have that:

$$h_{1,2} = \frac{122e}{10^7} < \frac{1}{1024} = h_{2,2}$$

Taking into account that $h_{1,i} = 122e \cdot \frac{t_{i-1}}{t_i} = 122e \cdot 2^{-\frac{t_{i-1}}{v_{\max}}}$ it becomes evident that h_1 drops to 0 faster than h_2 and thus the desired inequality is proved.

B.2 The Proof of Lemma 2

We are now ready to prove Lemma 2

Proof (Proof of Lemma 2:). First let R be the set of all the rounds that LM-Mechanism uses. Using lemma 1 we get the following trivial bound for the participation probability q_j of the j -th round:

$$q_j \geq a_j \cdot \prod_{i \in R} (1 - a_i)$$

All that remains is to bound $\prod_{i \in R} (1 - a_i)$. We break this product into 3 parts:

1. The first term of the product is $\frac{2}{3}$ and it corresponds to τ .
2. Next comes the following product of terms from the tests of the second period:

$$\prod_{i=2}^T (1 - a_i)^{\gamma_i} \geq \prod_{i=2}^T (1 - a_i \cdot \gamma_i),$$

where the last inequality comes from taking the first order Taylor approximation of the convex function $g(x) = (1 - x)^{\log(\frac{e}{x})}$. Using Lemma 11, we get:

$$\prod_{i=2}^T (1 - a_i)^{\gamma_i} \geq \prod_{i=2}^T \left(1 - \frac{1}{i^{10}}\right) \geq \frac{9}{10}.$$

3. Lastly, we have the terms from the BinarySearch and Exploitation:

$$\prod_{i=1}^{2 \cdot \ell} (1 - a) = (1 - a)^{2 \cdot \ell} \geq \frac{1}{e}.$$

Combining all of the above together:

$$q_j \geq a_j \cdot \prod_{i \in R} (1 - a_i) \geq \frac{6}{10e} \cdot a_j$$

Before we delve deeper into the properties of our rounds we first need to establish a basic property of Threshold-Based mechanisms.

B.3 The Proof of Lemma 3

Lemma 12 ([17]). *Let $k = \frac{\text{OPT}}{v_{\max}}$. For any $\hat{t} = h \cdot \text{OPT}$, such that $h \in (0, 1)$, the following holds:*

$$\text{PostedPrices}(\hat{t}, \mathcal{N}, B) \geq \min \left\{ (1 - h) \cdot \text{OPT}, \left(h - \frac{1}{k} \right) \cdot \text{OPT} \right\},$$

where $\text{PostedPrices}(\hat{t}, \mathcal{N}, B)$ denotes the expected value gathered by applying linear pricing with threshold \hat{t} to the set \mathcal{N} of agents with budget B .

Proof. Suppose that we use $\hat{t} = h \cdot \text{OPT}$ for the entire input. Let S be the allocation obtained using \hat{t} . We consider the following cases:

1. There is an agent b such that $b \in S^* \setminus S$ cause of insufficient Budget. In this case:

$$\begin{aligned} \sum_{b_j \in S} p_j &= \sum_{b_j \in S} \frac{B}{\hat{t}} (f_{S_j}(b_j)) = \frac{B \cdot f(S)}{\hat{t}} = B \cdot \frac{f(S)}{\text{OPT} \cdot h} \text{ and} \\ c(b) &\leq \frac{B}{\hat{t}} \cdot f(b) \leq B \cdot \frac{1}{\text{OPT} \cdot h} \cdot \frac{\text{OPT}}{k} = B \cdot \frac{1}{h \cdot k} \end{aligned}$$

and thus

$$\begin{aligned} B &< c(b) + \sum_{b_j \in S} p_j \leq B \cdot \frac{f(S)}{\text{OPT} \cdot h} + B \cdot \frac{1}{h \cdot k} \implies \\ \text{OPT} \cdot \left(h - \frac{1}{k} \right) &< f(S) \end{aligned}$$

2. Otherwise: $c(b) > \frac{B}{\hat{t}} \cdot f_S(b)$ for every $b \in S^* \setminus S$ and thus

$$\sum_{b \in S^* \setminus S} f_S(b) = \sum_{b \in S^* \setminus S} \frac{f_S(b)}{c(b)} \cdot c(b) \leq \sum_{b \in S^* \setminus S} \frac{\hat{t}}{B} \cdot c(b) \leq \hat{t}$$

and it follows that

$$\begin{aligned} \text{OPT} - f(S) &\leq f(S^* \cup S) - f(S) \leq \sum_{b \in S^* \setminus S} f_S(b) \leq \hat{t} \implies \\ f(S) &\geq (1 - h) \cdot \text{OPT} \end{aligned}$$

Finally, we conclude that $f(S) \geq \min\{(1 - h), (h - \frac{1}{k})\} \cdot \text{OPT}$.

We notice that the last bound is maximized for $\hat{h} = \frac{k+1}{2k}$ and thus we consider it to be exactly that for the rest of our analysis.

Let $\hat{t} = \hat{h} \cdot \text{OPT} = \frac{k+1}{2k} \cdot \text{OPT}$ and S be the allocation we get by applying the threshold \hat{t} to the whole input, for some permutation of the agents. Denote $\{s_i\}_{i=1}^{|S|}$ the elements of S in order of appearance and let $S_i = \bigcup_{j=1}^i \{s_j\}$. Now, letting $w_i = f_{S_{i-1}}(s_i)$, we can bound the cost of all subsets $X \subseteq S$ the following way:

Lemma 13. *For every subset $X \subseteq S$ it is true that:*

$$c(X) \leq \frac{\sum_{i: s_i \in X} w_i}{\hat{h} \cdot \text{OPT}} \cdot B$$

Proof. The cost of each agent in S is at most $\frac{f_{S_i}(s_i)}{t} \cdot B = \frac{w_i}{t} \cdot B$. Consequently, for each subset X of S we get:

$$c(X) = \sum_{j: b_j \in X} c(b_j) \leq \sum_{i: s_i \in X} \frac{w_i}{t} \cdot B.$$

The following lemma showcases a way to lower bound the value of subsets $X \subseteq S$.

Lemma 14. *Let I be the set of indices of elements $s_i \in S$ sampled in a round. Then, the following holds:*

$$f\left(\bigcup_{j \in I} \{s_j\}\right) \geq \sum_{j \in I} w_j$$

Proof. Let $J = \bigcup_{j \in I} \{s_j\}$. The value of the set J is the same regardless of the order of appearance of elements s_j . Thus we can suppose that they appear in the same order as in S and thus:

$$f(J) = \sum_{j \in I} f_{S_{j-1} \cap J}(s_j) \geq \sum_{j \in I} f_{S_{j-1}}(s_j) = \sum_{j \in I} w_j$$

Combining the lemmas 13 and 14 we get the following bound for the cost of all subsets $X \subseteq S$.

Corollary 1. *For every $X \subseteq S$ it holds that:*

$$c(X) \leq \frac{2k}{k+1} \cdot \frac{f(X)}{\text{OPT}} \cdot B$$

Let, once again, R be the set of rounds. Suppose that X is the subset of agents of S that is included in a round j with length parameter a_j , such that $a_j \cdot \frac{\text{OPT}}{v_{\max}} \geq 81e$. Now consider the following random variables for $j \in R, i \in [|S|]$:

$$X_i^j = \begin{cases} w_i & \text{w.p. } \frac{6}{10e} \cdot a_j \\ 0 & \text{otherwise} \end{cases}$$

$$X^j = \sum_{i=1}^{|S|} X_i^j$$

By lemmas 1, 14 it is evident that X^j is a lower bound of the value $f(X)$ contained in round j . Furthermore, we can prove the following:

Lemma 15. *Let OPT_j be the value of an optimal solution for an a_j -round j , when restricted to budget $B_j = 3 \cdot C \cdot a_j \cdot B$ budget, where a_j is good with respect to OPT . If $C \cdot a_j \cdot \text{OPT} \leq X^j$ then*

$$\text{OPT}_j \geq C \cdot a_j \cdot \text{OPT}$$

Proof. Let X be the set of agents included in round j . Define $S_j = X \cap S$ and denote by $s_{j,i}$ the i -th element of S_j in the order of appearance within the round. We can select the first μ agents $\{s_{j,i}\}_{i=1}^{\mu}$ (for some $\mu \in \mathbb{N}$) such that their total value $f(\{s_{j,i}\}_{i=1}^{\mu})$ does not exceed $C \cdot a_j \cdot \text{OPT}$ by more than v_{\max} . More specifically, there exists $\mu \in \mathbb{N}$ such that:

$$\begin{aligned} C \cdot a_j \cdot \text{OPT} &\leq f\left(\bigcup_{i=1}^{\mu} \{s_{j,i}\}\right) \\ &\leq C \cdot a_j \cdot \text{OPT} + v_{\max} \\ &= \left(C \cdot a_j + \frac{v_{\max}}{\text{OPT}}\right) \cdot \text{OPT} \end{aligned}$$

Which, by Corollary 1, implies:

$$c \left(\bigcup_{i=1}^{\mu} \{s_{j,i}\} \right) \leq 2 \cdot \left(C \cdot a_j \cdot + \frac{v_{\max}}{\text{OPT}} \right) \cdot B \leq 3 \cdot C \cdot a_j \cdot B,$$

where the last inequality is due to a_j being good with respect to OPT, i.e., $a_j \geq 81 e v_{\max} / \text{OPT}$.

Lemma 15 enables us to analyze the value of X^j to determine whether a round is dense, rather than relying on OPT_j . Now we can use concentration bounds on the random variable X_j to bound the probability that the j -th round is dense.

We now ready to conclude this section with presenting the proof of Lemma 3.

Proof (Proof of lemma 3:).

By Lemma 15:

$$\mathbb{P}[\text{round } j \text{ is not dense}] \leq \mathbb{P}[X^j < C \cdot a_j \cdot \text{OPT}]$$

We notice that $\frac{1}{2} \cdot \frac{6}{10e} \cdot \frac{k-1}{2k} \geq C$ (for $k \geq 10^7$) and thus:

$$\mathbb{P}[\text{round } j \text{ is not dense}] \leq \mathbb{P}\left[X^j < \frac{1}{2} \cdot \frac{6}{10e} \cdot a_j \cdot \frac{k-1}{2k} \cdot \text{OPT}\right]$$

Now using the fact that $\mathbb{E}[X^j] = q_j \cdot f(S)$, along with Lemma 1:

$$\mathbb{P}[\text{round } j \text{ is not dense}] \leq \mathbb{P}\left[X^j \leq \frac{1}{2} \cdot \mathbb{E}[X^j]\right]$$

We can now apply Chernoff bound (setting $\delta = \frac{1}{2}$) on random variable X^j to get:

$$\begin{aligned} \mathbb{P}[\text{round } j \text{ is not dense}] &\leq \mathbb{P}[X^j \leq (1 - \delta)\mathbb{E}[X^j]] \\ &\leq \exp\left\{\frac{-\delta^2 \mathbb{E}[X^j]}{2 \cdot \max X_i^j}\right\} \end{aligned}$$

Using the fact that $\mathbb{E}[X^j] = \frac{6}{10e} \cdot a_j \cdot f(S) \geq \frac{6}{10e} \cdot a_j \cdot \frac{k-1}{2k} \cdot \text{OPT}$:

$$\mathbb{P}[\text{round } j \text{ is not dense}] \leq \exp\left\{\frac{-\delta^2 \cdot \frac{6}{10e} \cdot a_j \cdot \frac{k-1}{2k} \cdot \text{OPT}}{2v_{\max}}\right\}$$

After calculations we reach the following:

$$\mathbb{P}[\text{round } j \text{ is not dense}] \leq \exp\left\{\frac{-a_j \cdot (k-1)}{27e}\right\} \leq 0.1$$

The last inequality follows from the fact that a_j is good with respect to OPT, i.e., $a_j \geq 81 e v_{\max} / \text{OPT}$, and from the hypothesis that $k \geq 10^7$.

B.4 The Proof of Lemma 4

To extend our analysis to dense phases we should first pay attention to the dependency between the probability of success of consecutive rounds. Indeed, we should notice that conditional to the previous round having failed the next round is more likely to succeed. Such a dependency bares a significant resemblance to the Balls and Bins problem [38]. Below we present how to utilize results from [39] to prove that concentration of mass arguments still hold in this setting.

In the following we will present a complete proof of the validity of Chernoff bound on the following random variables:

$$Q_i = \begin{cases} 1, & \text{if round } i \text{ is dense} \\ 0, & \text{otherwise} \end{cases}$$

Firstly, we need to present some important notions from [39].

Definition 1 (Negative Association). Let $\mathbf{X} := (X_1, \dots, X_n)$ be a vector of random variables.

(-A) The random variables, \mathbf{X} are negatively associated if for any two disjoint index sets, $I, J \subseteq [n]$,

$$\mathbb{E}[f(X_i, i \in I)g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)]\mathbb{E}[g(X_j, j \in J)]$$

for all functions $f : \mathbb{R}^{|I|} \rightarrow \mathbb{R}$ and $g : \mathbb{R}^{|J|} \rightarrow \mathbb{R}$ that are both non-decreasing or both non-increasing.

Lemma 16 (Zero-One Lemma for (-A)). If X_1, \dots, X_n are zero-one random variables such that $\sum_i X_i = 1$, then X_1, \dots, X_n satisfy (-A).

Proposition 4. 1. If \mathbf{X} and \mathbf{Y} satisfy (-A) and are mutually independent, then the augmented vector $(\mathbf{X}, \mathbf{Y}) = (X_1, \dots, X_n, Y_1, \dots, Y_m)$ satisfies (-A).

2. Let $\mathbf{X} := (X_1, \dots, X_n)$ satisfy (-A). Let $I_1, \dots, I_k \subseteq [n]$ be disjoint index sets, for some positive integer k . For $j \in [k]$, let $h_j : \mathbb{R}^{|I_j|} \rightarrow \mathbb{R}$ be functions that are all non-decreasing or all non-increasing, and define $Y_j := h_j(X_i, i \in I_j)$. Then the vector $\mathbf{Y} := (Y_1, \dots, Y_k)$ also satisfies (-A). That is, non-decreasing (or non-increasing) functions of disjoint subsets of negatively associated variables are also negatively associated.

Proposition 5. The Chernoff-Hoeffding bounds are applicable to sums of variables that satisfy the negative association condition (-A).

We are now ready to complete our proof:

Lemma 17. We can apply Chernoff bound on $\{Q_i\}_{i \in [\ell]}$

Proof. Consider the following random variables:

$$Y_i^j = \begin{cases} 1 & \text{if agent } b_i \text{ is in round } j \\ 0 & \text{otherwise} \end{cases}$$

$$Q_j = \begin{cases} 1 & \text{if } \sum_{i=1}^n w_i \cdot Y_i^j \geq C \cdot a_j \cdot OPT \\ 0 & \text{otherwise} \end{cases}$$

By lemma 16 it is immediate that $(Y_i^1, Y_i^2, \dots, Y_i^\ell)$ are negatively associated. By Proposition 4.1, along with the independency among agents, we can conclude that the whole set $(Y_i^j)_{i \in [n], j \in [\ell]}$ is negatively associated. Finally, we use proposition 4.2, setting

$$h_j = \begin{cases} 1 & \text{if } \sum_{i=1}^n w_i \cdot Y_i^j \geq C \cdot a \cdot OPT \\ 0 & \text{otherwise} \end{cases}$$

which is an increasing function of Y_i^j , to get that $(Q_1, Q_2, \dots, Q_\ell)$ are negatively associated. We now get the desired result directly from Proposition 5

The following lemma effectively leverages this dependency to demonstrate that all phases, whose rounds are good with respect to their thresholds, are dense with significant probability.

Proof (Proof of Lemma 4). Proposition 3 suggests that every round succeeds with probability at least 0.9. Now let $\{Q_i\}_{i=1}^\delta$ be random variables under the following distribution:

$$Q_i = \begin{cases} 1, & \text{if round } i \text{ is dense} \\ 0, & \text{otherwise} \end{cases}$$

By leveraging the previous results, we can argue that the variables Q_i exhibit negative dependence, which permits the use of concentration inequalities.

Let $Q = \sum_{i=1}^m Q_i$. The probability a Phase is not dense is, by applying Chernoff bound on Q , at most:

$$\begin{aligned} \mathbb{P}\left[Q < \frac{\delta}{2}\right] &\leq \mathbb{P}\left[Q < \frac{1}{1.8} \cdot \mathbb{E}[Q]\right] \\ &\leq \exp\left\{-\frac{\left(\frac{0.8}{1.8}\right)^2 \cdot 0.9 \cdot \delta}{2}\right\} \\ &= \exp\left\{-\frac{4 \cdot \delta}{45}\right\} \end{aligned}$$

B.5 Lemmas Used in Estimating the Correctness Probability in Section 4.3

Lemma 18. *The event \mathcal{E}_2 happens with probability at least 0.9.*

Proof. To successfully identify the correct intervals, we need the last, or the second to last, successful Phase to include $\frac{OPT}{8}$. Let us denote by P_i the Phase that tests threshold t_i . Let $\frac{OPT}{8} \in (t_j, t_{j+1}]$, then, if P_j is dense, P_j will be successful and all P_i , with $i > j + 1$ will fail, by Lemma 8. Thus, the set of intervals we end up with will include $(t_j, t_{j+1}]$. Thus, we need to investigate the probability that Phase P_j is dense. By Lemma 4 we have that:

$$\begin{aligned} \mathbb{P}[P_j \text{ is dense}] &\geq 1 - \exp\left\{-\frac{4 \cdot \gamma_j}{45}\right\} \\ &\geq 1 - \exp\left\{-\frac{4 \cdot \gamma_2}{45}\right\} = 1 - \exp\{-3\} \geq 0.9 \end{aligned}$$

Lemma 19. *All phases tested with a threshold $t \leq \frac{OPT}{8}$, during BinarySearch and Exploitation, are dense with probability at least 0.9 (i.e., the event \mathcal{E}_3 occurs with probability at least 0.9).*

Proof. In the worst case we might need all of the $\frac{2\ell}{m}$ Phases to be dense. Lemma 3 suggests that every round succeeds with probability at least 0.9. Now let $\{Q_i\}_{i=1}^m$ be random variables under the following distribution:

$$Q_i = \begin{cases} 1, & \text{if round } i \text{ is dense} \\ 0, & \text{otherwise} \end{cases}$$

Let $Q = \sum_{i=1}^m Q_i$. As argued before (B.4), we can apply concentration of mass arguments to Q . The probability a Phase is not dense is, by applying Chernoff bound on Q , at most:

$$\mathbb{P}\left[Q < \frac{m}{2}\right] \leq \mathbb{P}\left[Q < \frac{1}{1.8} \cdot \mathbb{E}[Q]\right] \leq \exp\left\{-\frac{(\frac{0.8}{1.8})^2 \cdot 0.9 \cdot m}{2}\right\}$$

Using the definitions of m and t_{\max} , we get:

$$\mathbb{P}\left[Q < \frac{m}{2}\right] \leq \frac{1}{\log\left(\frac{t_{\max}}{t_{\min}}\right)} = \frac{1}{t_{\min}}$$

Leveraging our hypothesis that $t_{\min} \geq 10^7$:

$$\mathbb{P}\left[Q < \frac{m}{2}\right] \leq \frac{1}{20 \lceil \log(t_{\min}) \rceil} \quad (8)$$

To get the desired probability bound, we will now use Union Bound on the event where the first $\frac{2\ell}{m}$ Phases are dense. Namely:

$$\mathbb{P}\left[\text{First } \frac{2 \cdot \ell}{m} \text{ Phases are dense}\right] \geq 1 - 2 \cdot \left\lceil \log \log \left(\frac{t_{\max}}{t_{\min}}\right) \right\rceil \cdot \mathbb{P}[\text{A Phase is not dense}]$$

Now using the definition of $t_{\max} = 2^{\frac{t_{\min}}{v_{\max}}} \cdot t_{\min}$, along with inequality 8:

$$\mathbb{P}\left[\text{First } \frac{2 \cdot \ell}{m} \text{ Phases are dense}\right] \geq 1 - \frac{\lceil \log(t_{\min}) \rceil}{10 \cdot \lceil \log(t_{\min}) \rceil} = 0.9$$

Sampling round lengths from a distribution introduces the risk that we run out of agents during the execution of our mechanism. Consequently, our analysis relies on the assumption that the total round lengths remain within the input size. Fortunately, the following lemma guarantees that this condition holds with constant probability:

Lemma 20. *We will not run out of agents during the execution of LM-Mechanism with probability at least 0.97 (i.e., the event \mathcal{E}_4 occurs with probability at least 0.97).*

Proof. We will investigate the round lengths drawn as follows:

1. Applying Chernoff bound onto the random length τ drawn for the estimation of v_{\max} we get that:

$$\mathbb{P}\left[\tau \geq \frac{2}{5} \cdot n\right] \leq \exp\left\{-\frac{n}{225}\right\} \leq 0.01,$$

where the last inequality holds for $n \geq 10^7$, which must be true for our assumption $\text{OPT} \geq 10^7 v_{\max}$ to hold. Thus:

$$\mathbb{P}\left[\tau < \frac{2}{5} \cdot n\right] \geq 0.99$$

2. We now consider the random lengths $n_{i,j}$ drawn during Period 2. Consider $m_{i,j} \sim B(n, a_i)$. If we were to use m_i as our round lengths then our risk of running out of agents would be greater than with n_i . We observe that:

$$\mathbb{E} \left[\sum_{i=2}^T \sum_{j=1}^{\gamma_i} n_{i,j} \right] \leq \mathbb{E} \left[\sum_{i=2}^T \sum_{j=1}^{\gamma_i} m_{i,j} \right] = n \cdot \sum_{i=2}^T \gamma_i \cdot a_i$$

By applying Markov's inequality on the sum of these random variables, and noticing that $\mathbb{E}[m_{i,j}] = n \cdot a_i$, we get:

$$\mathbb{P} \left[\sum_{i=2}^T \sum_{j=1}^{\gamma_i} n_{i,j} \geq 100 \cdot n \cdot \sum_{i=2}^T a_i \cdot \gamma_i \right] \leq \mathbb{P} \left[\sum_{i=2}^T \sum_{j=1}^{\gamma_i} m_{i,j} \geq 100 \cdot n \cdot \sum_{i=2}^T a_i \cdot \gamma_i \right] \leq 0.01$$

Thus:

$$\mathbb{P} \left[\sum_{i=2}^T \sum_{j=1}^{\gamma_i} n_{i,j} \geq 100 \cdot n \cdot \sum_{i=2}^T a_i \cdot \gamma_i \right] \leq 0.01$$

By lemma 11 we get $\sum_{i=2}^T a_i \cdot \gamma_i < \frac{1}{500}$, and thus:

$$100 \cdot n \cdot \sum_{i=2}^T a_i \cdot \gamma_i < \frac{n}{5}$$

Overall:

$$\mathbb{P} \left[\sum_{i=2}^T \sum_{j=1}^{\gamma_i} n_{i,j} < \frac{n}{5} \right] \geq 0.99$$

3. Finally, we draw $2 \cdot \ell$ lengths $n_i \sim B(N_i, a)$ for the rounds of BinarySearch and Exploitation with the same length parameter a . Suppose, similarly with before, the random variables $m_i \sim B(n, a)$ for $i \in [2\ell]$. It is evident that

$$\mathbb{E} \left[\sum_{i=1}^{2\ell} n_i \right] \leq \mathbb{E} \left[\sum_{i=1}^{2\ell} m_i \right] = n \cdot 2\ell \cdot a = \frac{n}{3}.$$

By applying now Chernoff bound on m_i we get that:

$$\mathbb{P} \left[\sum_{i=1}^{2\ell} n_i \geq \frac{2}{5} \cdot n \right] \leq \mathbb{P} \left[\sum_{i=1}^{2\ell} m_i \geq \frac{2}{5} \cdot n \right] \leq 0.01.$$

Overall we get:

$$\mathbb{P} \left[\sum_{i=1}^{2\ell} n_i < \frac{2}{5} \cdot n \right] \geq 0.99$$

All in all, by Union Bound on the events described before, we get that we will not run out of agents during our Mechanism with probability at least $1 - 0.01 - 0.01 - 0.01 = 0.97$

C The Proof of Lemma 6

Below we prove that all needed rounds can be conducted without fear of running out of budget.

Proof (Proof of Lemma 6:). According to Property 1 during Period 2 in worst case the overall budget expended is the following:

$$\sum_{i=2}^T \gamma_i \cdot 3 \cdot C \cdot a_i \cdot B \leq \sum_{i=2}^T \frac{3 \cdot C}{i^{10}} \cdot B < 0.04 \cdot B,$$

where the second to last inequality follows from Lemma 11.

According to Property 1, during Period 3 and 4, in worst case the overall budget expended is the following:

$$\begin{aligned} \sum_{i=1}^{2 \cdot \ell} 3 \cdot C \cdot a \cdot B &\leq 6 \cdot C \cdot \ell \cdot a \cdot B \\ &= C \cdot B \\ &\leq 0.06 \cdot B \end{aligned}$$

Overall we have expended a budget of at most $B/10$.