A Unifying Framework for Online Optimization with Long-Term Constraints

Matteo Castiglioni∗
Politecnico di Milano

Andrea Celli†
Bocconi University

Alberto Marchesi∗
Politecnico di Milano

Giulia Romano∗
Politecnico di Milano

Nicola Gatti∗
Politecnico di Milano

Abstract

We study online learning problems in which a decision maker has to take a sequence of decisions subject to \( m \) long-term constraints. The goal of the decision maker is to maximize their total reward, while at the same time achieving small cumulative constraints violation across the \( T \) rounds. We present the first best-of-both-world type algorithm for this general class of problems, with no-regret guarantees both in the case in which rewards and constraints are selected according to an unknown stochastic model, and in the case in which they are selected at each round by an adversary. Our algorithm is the first to provide guarantees in the adversarial setting with respect to the optimal fixed strategy that satisfies the long-term constraints. In particular, it guarantees a \( \rho/(1 + \rho) \) fraction of the optimal reward and sublinear regret, where \( \rho \) is a feasibility parameter related to the existence of strictly feasible solutions. Our framework employs traditional regret minimizers as black-box components. Therefore, by instantiating it with an appropriate choice of regret minimizers it can handle the full-feedback as well as the bandit-feedback setting. Moreover, it allows the decision maker to seamlessly handle scenarios with non-convex rewards and constraints. We show how our framework can be applied in the context of budget-management mechanisms for repeated auctions in order to guarantee long-term constraints that are not packing (e.g., ROI constraints).

1 Introduction

We study online learning problems where a decision maker takes decisions over \( T \) rounds. At each round \( t \), the decision \( x_t \in \mathcal{X} \) is chosen before observing a reward function \( f_t \) together with a set of \( m \) time-varying constraint functions \( g_t \). The decision maker is allowed to make decisions that are not feasible, provided that the overall sequence of decisions obeys the long-term constraints

\[
\sum_{t=1}^{T} g_t(x_t) \leq 0,
\]

up to a small cumulative violation across the \( T \) rounds. The problem becomes that of finding a sequence of decisions \( x_t \) which guarantees a reward close to that of the best fixed decision in hindsight while satisfying long-term constraints. This type of framework was first proposed by Mannor et al. [36], and it has numerous applications ranging from wireless communication [36] and multi-objective online classification [13], to safe online learning [4].

Mannor et al. [36] show that guaranteeing sublinear regret and sublinear cumulative constraints violation is impossible even when \( f_t \) and \( g_t \) are simple linear functions. Therefore, previous works either focus on the case in which constraints are generated i.i.d. according to some unknown stochastic

∗DEIB, Politecnico di Milano, {matteo.castiglioni, alberto.marchesi, giulia.romano, nicola.gatti}@polimi.it.
† Computing Sciences Department, Bocconi University, andrea.celli2@unibocconi.it.

model, without providing any guarantees for the adversarial case, or provide results for adversarially-generated constraints under some strong assumptions on the structure of the problem or using a weaker baseline (a detailed discussion of related works can be found in Appendix A). A few examples in the latter case are [16, 21, 40, 42]. In the former setting (i.e., stochastic constraints), Wei et al. [41] consider a weaker baseline that is feasible for each constraint \( g_t \), going against the basic idea of long-term constraints. A notable exception is the work by Yu et al. [44], who employ the same baseline as ours, and provide an upper bound of \( \tilde{O}(T^{1/2}) \) for both regret and constraints violation (see Table 1). We also mention that there are some works studying the problem in which constraints are static (see, e.g., [31, 35, 43, 45]), or focus on specific types of constraints, such as knapsack constraints [7, 30]. Our framework differs from those works as we deal with arbitrary and time-varying constraints. Moreover, it also extends the online convex optimization framework introduced by Zinkevich [46] by allowing for general non-convex loss functions \( f_t \), arbitrary feasibility sets \( \mathcal{X} \), and arbitrary time-varying long-term constraints.

### 1.1 Original contributions

Given the negative result by Mannor et al. [36], a natural question is what kind of guarantees we can reach in the adversarial setting, when adopting the standard baseline of the best fixed decision in hindsight satisfying (in expectation) the long-term constraints. We provide the first positive result going in this direction, by designing a no-\( \alpha \)-regret algorithm that guarantees a sublinear cumulative constraints violation. Moreover, we make a step forward in the line of work initiated by Bubeck and Slivkins [13], by showing that our algorithm is also the first best-of-both-worlds algorithm for problems with arbitrary long-term constraints. This allows our algorithm to guarantee good worst-case performance (adversarial case), while being able to exploit well-behaved problem instances (stochastic case). The only assumption which we require is the existence of a decision that is strictly feasible (see Section 2 for a definition). At the same time, we show that even without this assumption, we can recover sublinear regret and violation with stochastic constraints.

Previous work usually assumes that \( \rho \) is a given constant. In that case, our algorithm matches the guarantees by Yu et al. [44] when constraints are generated i.i.d. according to an unknown distribution, and has no-\( \alpha \)-regret with \( \alpha = \rho / (1 + \rho) \) in the adversarial case (see Table 1). Our algorithm only requires a lower bound on the real value of the feasibility parameter \( \rho \). In the stochastic case, the lower bound may even be unknown, and the algorithm can efficiently estimate it from data. Moreover, we argue that if \( \rho \) is allowed to depend on \( T \) and take arbitrarily small values, then there are certain values \( \rho \leq T^{-1/4} \), for which any regret bound depending on \( 1/\rho \) would be useless (i.e., not sublinear in \( T \), see Section 3). This setting is usually overlooked by previous work, which assumes \( \rho \) to be a given constant. We show that, in the case of an arbitrary feasibility parameter \( \rho \), in the stochastic setting our algorithm guarantees an upper bound of \( \tilde{O}(T^{3/4}) \) for regret and cumulative constraints violation.

Our framework employs traditional regret minimizers as black-box components. Therefore, by instantiating it with an appropriate choice of regret minimizers it can handle full-feedback as well as bandit-feedback settings. In the former case, after playing \( x_t \), the decision maker gets to observe \( f_t \) and \( g_t \), while in the latter case only the realized values \( f_t(x_t) \) and \( g_t(x_t) \) are observed. Moreover, this allows the decision maker to seamlessly handle scenarios with non-convex reward and constraints, by employing a suitable regret minimizer for non-convex losses (see, e.g., [39]). Our algorithm is based on a two-stage approach in which primal and dual players interact through Lagrangian games.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Constr.</th>
<th>Non-convex ( f_t ) and ( g_t )</th>
<th>Bound — constant ( \rho )</th>
<th>Bound — arbitrary ( \rho )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Reward</td>
<td>Violation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \tilde{O}(T^{1/2}) )</td>
<td>( \tilde{O}(T^{1/2}) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \tilde{O}(T^{1/2}) )</td>
<td>( \tilde{O}(T^{1/2}) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \tilde{O}(T^{3/4}) )</td>
<td>( \tilde{O}(T^{3/4}) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Comparison between the performance of our algorithm and previous work using the same baseline as ours. Bounds for settings that were not previously tractable are highlighted in gray. \( \text{OPT} \) is the reward of the baseline.
In the first (play) phase, the primal player tries to balance out the maximization of their rewards with constraints violation. In the second (recovery) phase, the primal player only makes “safe decisions” to avoid violating constraints too much. It is possible to prove that, in the case of stochastic rewards and constraints, the algorithm never enters phase two. This property is particularly relevant for budget-pacing mechanisms in repeated auctions, since it is related to how budget is allocated. Our framework can also be instantiated to perform budget allocation subject to constraints that were previously not tractable by traditional mechanisms, such as ROI constraints \[11, 22\].

2 Preliminaries

The decision maker has a non-empty set of available strategies \( X \) (this set may be non-convex, integral, and even non-compact). In each round \( t \in [T] \), the decision maker first chooses \( x_t \in X \), and the environment selects a reward function \( f_t : X \to [0, 1] \) and a constraint function \( g_t : X \to [-1, 1]^m \) conditioned on the past history of play up to time \( t-1 \) (i.e., the environment chooses \( f_t \) and \( g_t \) without knowledge of \( x_t \)). Notice that both \( f_t \) and \( g_t \) need not be convex. The latter specifies a set of \( m \) constraints of the form \( g_t(x) \leq 0 \), with \( g_t,i(x) \leq 0 \) denoting the \( i \)-th constraint.\(^3\) In the following, we denote as \( \mathcal{F} \), respectively \( \mathcal{G} \), the set of all the possible \( f_t \), respectively \( g_t \), functions (e.g., \( \mathcal{F} \) and \( \mathcal{G} \) may contain all the Lipschitz-continuous functions defined over \( X \)). At each round \( t \in [T] \), the decision maker can condition their decision on prior feedbacks and on the sequence of prior decisions \( x_1,\ldots,x_{t-1} \), but no information about future rewards and constraint functions is available.

2.1 Strong duality through strategy mixtures

In the following, we define the optimization problem (Problem \([\text{LP}_{f,g}]\)) which is used to define the probabilistic baselines against which we compare the performances of the decision maker. Such a problem involves probabilistic mixtures of strategies in \( X \), which are crucial in order to recover strong duality.\(^4\)

First, we introduce the set of probability measures on the Borel sets of \( X \). We refer to such a set as the set of strategy mixtures, denoted as \( \Xi \). We endow \( X \) with the Lebesgue \( \sigma \)-algebra, and we assume that all the functions in \( \mathcal{F} \) and \( \mathcal{G} \) are measurable with respect to every probability measure \( \xi \in \Xi \). This ensures that the various expectations taken are well-defined, since the functions are assumed to be bounded above, and they are therefore integrable. In the following, for ease of presentation and with a slight abuse of notation, whenever we write a \( \xi \in \Xi \) in place of an \( x \in X \), we mean that we are taking the expectation with respect to the probability measure \( \xi \). For instance, given \( f \in \mathcal{F} \) and \( g \in \mathcal{G} \), we have that \( f(\xi) = \mathbb{E}_{x \sim \xi} f(x) \) and \( g(\xi) = \mathbb{E}_{x \sim \xi} g(x) \).

Then, given two functions \( f \in \mathcal{F} \) and \( g \in \mathcal{G} \), we define the following optimization problem, which chooses the strategy mixture \( \xi \in \Xi \) that maximizes the expected reward encoded by \( f \), while guaranteeing that the constraints encoded by \( g \) are satisfied in expectation.

\[
\text{OPT}_{f,g} := \left\{ \sup_{\xi \in \Xi} f(\xi) \quad \text{s.t.} \quad g(\xi) \leq 0 \right\} \quad (\text{LP}_{f,g})
\]

We denote by \( d_g \in [-1, 1] \) the largest possible value for which there exists a strategy mixture \( \xi \in \Xi \) satisfying the constraints \( g(\xi) \leq 0 \) by a margin of at least \( d_g \). Formally,

\[
d_g := \sup_{\xi \in \Xi} \min_{i \in [m]} -g_i(\xi). \tag{1}
\]

In order to ensure that \( \text{OPT}_{f,g} \) is always well defined, we assume that it is always the case that \( d_g \geq 0 \). Notice that, if \( d_g > 0 \), then Problem \([\text{LP}_{f,g}]\) satisfies Slater’s condition.

In the following, we prove some auxiliary results relating to Problem \([\text{LP}_{f,g}]\) that will be useful in the rest of the paper. First, we introduce a Lagrangian relaxation of the problem.

\(^2\)In this work, we denote by \( [x] \) the set \( \{1,\ldots,x\} \) of the first \( x \) natural numbers.

\(^3\)Focusing on the case \( g_t(x) \leq 0 \) is w.l.o.g. since any set of constraints can be represented in such a form.

\(^4\)The optimal fixed strategy mixture provides an arguably stronger baseline than the optimal fixed strategy. In stochastic settings, this baseline is related to the best dynamic policy. In particular, if we consider the case in which the observed functions are defined as the average of functions \( f_t \) and \( g_t \) across the \( T \) rounds, then the optimal mixture provides the same utility as the best dynamic policy (see \[2\] for a similar result).
**Definition 2.1 (Lagrangian Function).** Given two arbitrary functions \( f \in \mathcal{F} \) and \( g \in \mathcal{G} \), the Lagrangian function \( L_{f,g} : \Xi \times \mathbb{R}_{\geq 0}^m \to \mathbb{R} \) of Problem \([\text{LP}_{f,g}]\) is defined as
\[
L_{f,g}(\xi, \lambda) := f(\xi) - \langle \lambda, g(\xi) \rangle.
\]

If Problem \([\text{LP}_{f,g}]\) satisfies Slater’s condition, then Theorem 1 of Chapter 8.3 in [34] readily gives us that strong duality holds even if \( f \) and \( g \) are arbitrary non-convex functions. Formally:

**Corollary 2.2.** Given \( f \in \mathcal{F} \) and \( g \in \mathcal{G} \) such that \( d_g > 0 \), it holds
\[
\sup_{\xi \in \Xi} \inf_{\lambda \in \mathbb{R}_{\geq 0}^m} L_{f,g}(\xi, \lambda) = \inf_{\lambda \in \mathbb{R}_{\geq 0}^m} \sup_{\xi \in \Xi} L_{f,g}(\xi, \lambda) = \OPT_{f,g}.
\]

Next, we show that, if \( d_g > 0 \), then strong duality holds even when we restrict the admissible dual vectors \( \lambda \in \mathbb{R}_{\geq 0}^m \) to the set \( \mathcal{D}_g \), where, for any \( q \in \mathbb{R}_{>0} \), we let \( \mathcal{D}_q := \{ \lambda \in \mathbb{R}_{\geq 0}^m : \| \lambda \|_1 \leq 1/q \} \) (omitted proofs can be found in Appendix B).

**Theorem 2.3.** Given \( f \in \mathcal{F} \) and \( g \in \mathcal{G} \) such that \( d_g > 0 \), it holds
\[
\sup_{\xi \in \Xi} \inf_{\lambda \in \mathcal{D}_q} L_{f,g}(\xi, \lambda) = \inf_{\lambda \in \mathcal{D}_q} \sup_{\xi \in \Xi} L_{f,g}(\xi, \lambda) = \OPT_{f,g}.
\]

### 2.2 Stochastic vs. adversarial: baselines and feasibility

We consider several settings, differing in how functions \( f_t \) and \( g_t \) are selected, either *stochastically* or *adversarially*. We say that functions \( f_t \) (respectively \( g_t \)) are selected stochastically, when they are independently drawn according to a given probability measure \( \mu_F \) over \( \mathcal{F} \) (respectively \( \mu_G \) over \( \mathcal{G} \)). Instead, we say that functions \( f_t \) (respectively \( g_t \)) are selected adversarially if each \( f_t \) (respectively \( g_t \)) is chosen by an adversary based on the sequence of prior decisions, namely \( x_1, \ldots, x_{t-1} \).

Consistently with previous work (see, e.g., [36]), we compare the performance of the decision maker (in terms of reward cumulated over the \( T \) rounds) against the baseline \( T \OPT_{f,g} \) (as defined by Problem \([\text{LP}_{f,g}]\)), where \( f \) and \( g \) are suitably-defined functions. In particular:

- When functions \( f_t \), respectively \( g_t \), are selected stochastically, then we define function \( \bar{f} \), respectively \( \bar{g} \), so that \( f(x) := \mathbb{E}_{f \sim \mu_F}[f(x)] \), respectively \( g(x) := \mathbb{E}_{g \sim \mu_G}[g(x)] \).

- When functions \( f_t \), respectively \( g_t \), are selected adversarially, then we define function \( \bar{f} \), respectively \( \bar{g} \), so that \( f(x) := \frac{1}{T} \sum_{t=1}^T f_t(x) \), respectively \( g(x) := \frac{1}{T} \sum_{t=1}^T g_t(x) \).

Intuitively, in the stochastic case, the baseline is instantiated with an expectation of functions taken with respect to the probability measure \( \mu_F \) (respectively \( \mu_G \)). Instead, in the adversarial case, the baseline uses the average of functions \( f_t \) (respectively \( g_t \)) observed over the \( T \) rounds.

Let us remark that, when the set \( \mathcal{X} \) is compact convex and functions \( f_t \) and \( g_t \) are convex, then Problem \([\text{LP}_{f,g}]\) defining our baselines can be equivalently re-written by using strategies \( x \in \mathcal{X} \) rather than strategy mixtures \( \xi \in \Xi \), since there always exists an optimal solution to Problem \([\text{LP}_{f,g}]\) that places all the probability mass on a single strategy.

Our goal is to design online algorithms for the decision maker that output a sequence of decisions \( x_1, \ldots, x_T \) such that both the cumulative regret with respect to the performance of the baseline, defined as \( R_T := T \OPT_{f,g} - \sum_{t=1}^T f_t(x_t) \), and the cumulative constraints violation, defined as \( V_T := \max_{i \in [m]} \sum_{t=1}^T g_{t,i}(x_t) \), grow sublinearly in the number of rounds \( T \).

In conclusion, we introduce a problem-specific parameter that is strictly related to the feasibility of Problem \([\text{LP}_{f,g}]\). We call it the feasibility parameter \( \rho \in \mathbb{R} \), which is formally defined as follows:

- When functions \( g_t \) are selected stochastically, \( \rho := \sup_{\xi \in \Xi} \min_{i \in [m]} - \bar{g}_{t,i}(\xi) \).

- When functions \( g_t \) are selected adversarially, \( \rho := \sup_{\xi \in \Xi} \min_{t \in [T]} \min_{i \in [m]} - g_{t,i}(\xi) \).

Intuitively, in the stochastic case, \( \rho \) is equal to \( d_g \), while in the adversarial case it is computed similarly, but considering the worst case with respect to the functions \( g_t \) observed at each round \( t \). Notice that, when \( \rho > 0 \), Slater’s condition is satisfied for Problem \([\text{LP}_{f,g}]\).
In the following, we denote by $\xi^* \in \Xi$ a strategy mixture that is optimal for Problem $\text{LP}_{f,g}$. Moreover, we always assume that functions $f_t$ and $g_t$ are such that Problem $\text{LP}_{f,g}$ is feasible, and we let $\xi^* \in \Xi$ be the feasible strategy mixture that is optimal for the problem defining $\rho^*$.

2.3 Regret minimizers

A regret minimizer (RM) for a set $\mathcal{W}$ is an abstract model for a decision maker repeatedly interacting with an environment. At each $t$, a RM performs two operations: (i) NEXTELEMENT($\cdot$), which outputs an element $w_t \in \mathcal{W}$; and (ii) OBSERVEUTILITY($\cdot$), which updates the internal state of the RM using the feedback received from the environment. This is either a utility function $u_t : \mathcal{W} \rightarrow [a, b]$ having range $[a, b] \subseteq \mathbb{R}$ (full feedback) or only the value $u_t(w_t)$ (bandit feedback), with $u_t$ possibly depending adversarially on $w_1, \ldots, w_{t-1}$. The objective of the RM is to output a sequence $w_1, \ldots, w_T$ of points in $\mathcal{W}$ so that its cumulative regret, defined as $\sup_{w \in \mathcal{W}} \sum_{t=1}^T (u_t(w) - u_t(w_t))$, grows asymptotically sublinearly in $T$. See [19] for a review of the various RMs available in the literature.

For the ease of presentation, we introduce the concept of regret minimizer constructor, which is a procedure, say INIT($\mathcal{W}, [a, b], \eta$), that builds a RM on the basis of the three parameters given as input. In particular, the procedure returns a RM instantiated for the set $\mathcal{W}$, working with utility functions having range $[a, b]$, and such that its cumulative regret is guaranteed to grow sublinearly in the time horizon $T$ with probability at least $1 - \eta$.

3 A unifying meta-algorithm

In this section, we present our meta-algorithm. Its core idea is to instantiate suitable pairs of RMs, where one is working in the domain $\mathcal{X}$ of primal variables and the other in a suitable subset of the domain $\mathbb{R}_+^m$ of dual variables. At each round $t \in [T]$, the algorithm makes the RMs “play” against each other in a Lagrangian game, where the utility functions observed by them are related to the Lagrangian function $L_{f_t,g_t}(x, \lambda)$ of Problem $\text{LP}_{f,g}$.

Algorithm [1] provides the pseudo-code of the meta-algorithm, which takes as input: the total number of rounds $T$, a failure probability $\delta \in (0, 1)$ such that the guarantees provided by the algorithm hold with probability at least $1 - \delta$, and a lower bound $\rho \geq 0$ on the value of the feasibility parameter $\rho$.

Algorithm description. The algorithm works in two phases. In the first one, called play phase, the algorithm builds a primal RM, called $R^p_\rho$, working in the primal domain $\mathcal{X}$ and a dual RM, called $R^d_\rho$, operating on the subset $D_\rho$ of the dual domain $\mathbb{R}_+^m$, where $\rho$ is set in Line 1. The algorithm makes the two RMs playing against each other (see the call LAGRANGIANGAME$(R^p_\rho, R^d_\rho, 1)$) until either the cumulative violation $V^\delta$ incurred by the algorithm exceeds a given threshold (see Line 4) where $M_\rho$ is defined in Equation (2) or round $T$ is reached. Then, in the second phase, called recovery phase, the algorithm constructs a new pair of primal, dual RMs, with the latter working on the $(m - 1)$-dimensional simplex $\Delta_m$. The recovery phase uses the remaining rounds to make these new RMs play against each other, with the primal RM observing modified utility functions that do not account for functions $f_t$ (see the call LAGRANGIANGAME$(R^p_{\rho}, R^d_{\rho}, 0)$). Intuitively, this is needed in order to ensure that the algorithm plays strategies $x_t$ that satisfy the constraints, thus balancing out the cumulative constraint violation accumulated in the first phase. The pseudo-code describing one “play” between two RMs, called $R^p$ and $R^d$, is defined by the sub-procedure LAGRANGIANGAME$\left(R^p, R^d, v\right)$ in Algorithm [2]. The additional parameter $v \in \{0, 1\}$ is used to control the feedback fed into the primal RM $R^p$: specifically, if $v = 1$, then $R^d$ observes a utility function that also accounts for $f_t$ (play phase), otherwise, if $v = 0$, the observed utility function only accounts for the term depending on $g_t$ (recovery phase).

Regret minimizer constructors. Algorithm [1] also needs access to two suitably-defined regret minimizer constructors, namely INIT$^p(\mathcal{W}, [a, b], \eta)$ and INIT$^d(\mathcal{W}, [a, b], \eta)$, where the former is used

\footnote{Notice that $\xi^*$ and $\xi^\circ$ may not be well defined in all the cases in which the problem that defines them does not admit a maximum. Nevertheless, in such cases, we assume that $\xi^*$ (or $\xi^\circ$) is a strategy mixture arbitrarily “close” to the supremum, so that all of our results continue to hold up to negligible additive approximations that are dominated by other approximation factors, and we can safely ignore them for ease of exposition.}

\footnote{The idea of having pairs of primal, dual RMs playing a Lagrangian game was originally introduced by Immorlica et al. [30], restricted to the case of knapsack constraints.}
to build RMIs working in the primal domain and the latter for those operating on the dual one. Their actual implementation depends on the specific problem at hand. In the following, we let $E_{t,n}^{\rho}$ be the regret upper bound (on $t \in [T]$ rounds) for primal RMs $\mathcal{R}^p$ dealing with utility functions having range $[0,1]$, as returned by the call $\text{INIT}^p(\mathcal{X},[0,1],\eta)$. Notice that, when the range is $[a,b]$, the same RM can be adopted by first normalizing utility values, so that the resulting regret upper bound is $(b-a)\mathcal{E}_{t,n}^{\rho}$. As for dual RMs $\mathcal{R}^d$, we let $\mathcal{E}_{t,n}^{\rho}$ be the regret upper bound (on $t \in [T]$ rounds) provided by the RM defined for the set $\Delta_m$, while $\mathcal{E}_{t,n}^{\rho}$ is the upper bound for the dual RM instantiated on the set $D_p$. Notice that, since dual RMs always have full feedback, we can safely assume that the regret bounds $\mathcal{E}_{t,n}^{\rho}$ hold deterministically. We also assume that RMs provide bounds that increase with the number of rounds, i.e., such that $\mathcal{E}_{t,n}^{\rho} \leq \mathcal{E}^{\rho}_{t',n}$ for all $t \leq t'$.

**How to construct RMs.** $\text{INIT}^p$ can be implemented by using online mirror descent (OMD) with domain $\Delta_m$ (or $D_1$) and a negative entropy regularizer. Since the utility function $u_t^p$ is linear in $\lambda$, we get a regret bound for the primal RM of $\mathcal{E}_{T,n}^{\rho} = O(\sqrt{T \log(m)})$ (see, e.g., [12, 37]). The design of $\text{INIT}^d$ depends on the structure of $\mathcal{X}$ and functions $f_t$ and $g_t$. For instance, in convex settings with full feedback we can employ OMD [29], while with bandit feedback we can use [15]. Finally, for non-convex functions we can employ, e.g., the RMs in [39]. All these RMs guarantee $O(\sqrt{T})$ regret.

**How to get away with no knowledge of $\rho$.** In Section 7, we show that a lower bound $\hat{\rho}$ is not necessary when functions $g_t$ are selected stochastically. Indeed, it is sufficient to add a preliminary phase to Algorithm 1 which is used to infer a suitable lower bound on $\rho$ from experience. In order to do this, only $\sqrt{T}$ rounds are needed, so that the bounds on the cumulative regret and the cumulative constraint violation achieved by the algorithm are not compromised. When functions $g_t$ are chosen adversarially, as we show in Section 7 it is impossible to compute a lower bound on the feasibility parameter $\rho$ by only adding a preliminary phase to Algorithm 1 which uses $\sqrt{T}$ rounds.

**Remark 3.1** (Dependence on the lower bound $\hat{\rho}$). Algorithm 1 can take as input any $\hat{\rho} \geq 0$. However, since our regret bounds include a factor $1/\hat{\rho}$, by choosing the trivial lower bound $\hat{\rho} = 0$ we incur in a regret of $O(\sqrt{T}/\hat{\rho}) = O(T^{3/4})$. In order to obtain optimal bounds, we would like to have $\hat{\rho} = \Omega(\rho)$.

**Remark 3.2** (Dependence on the feasibility parameter $\rho$). In our analysis, we include the dependence on the feasibility parameter $\rho$ in the regret and constraint violation bounds achieved by the algorithm. As customary, the goal is devising bounds of the form $\text{poly}(\text{instance}) \cdot b(T)$, where the first term is a polynomial function of the parameters defining the problem instance, and $b(T) = o(T)$. Therefore, we cannot include a factor $1/\rho$ in the regret bounds if $\rho$ can be arbitrarily small. Even from a practical standpoint, when $\rho$ is too small, a $1/\rho$ regret bound is too large to be significant. For those reasons, we set $\hat{\rho}$ in Algorithm 1 to be the maximum between the feasibility parameter lower bound $\hat{\rho}$ and $T^{-1/4}$. Notice that the value $T^{-1/4}$ has been carefully chosen so as to minimize the maximum between the cumulative regret and the cumulative constraint violation when the lower bound on the feasibility parameter $\hat{\rho}$ is too small.

---

**Algorithm 1** META-ALGORITHM($T, \delta, \hat{\rho}$)

1: $\hat{\rho} \leftarrow \max \left\{ \hat{\rho}/2, T^{-1/4} \right\}$, $\eta \leftarrow \delta/3$, $t \leftarrow 1$
   \> Phase I: Play
2: $\mathcal{R}^p_t \leftarrow \text{INIT}^p(\mathcal{X},[1-1/\hat{\rho}, 1+1/\hat{\rho}], \eta)$
3: $\mathcal{R}^d_t \leftarrow \text{INIT}^d(D, [1-1/\hat{\rho}, 1/\hat{\rho}], 0)$
4: while $t \leq T$ do
5:   \> Phase II: Recovery
6:   $T_i \leftarrow t - 1$
7:   $\mathcal{R}^p_{t,n} \leftarrow \text{INIT}^p(\mathcal{X}, [-1, 1], \eta)$
8:   $\mathcal{R}^d_{t,n} \leftarrow \text{INIT}^d(D_m, [-1, 1], 0)$
9: while $t \leq T$ do
10:   $\mathcal{R}^p_{t,n} \leftarrow \text{LAGRANGIANGAME}(\mathcal{R}^p_t, \mathcal{R}^d_t, 1)$
11:   $\mathcal{R}^d_{t,n} \leftarrow \text{LAGRANGIANGAME}(\mathcal{R}^p_t, \mathcal{R}^d_t, 0)$
12:   $t \leftarrow t + 1$

---

**Algorithm 2** LAGRANGIANGAME($\mathcal{R}^p$, $\mathcal{R}^d$, $\upsilon$)

1: $\mathbf{x}_t \leftarrow \mathcal{R}^p_.\text{NEXTELEMENT}()$
2: $\mathbf{A}_t \leftarrow \mathcal{R}^d_.\text{NEXTELEMENT}()$
3: Play $\mathbf{x}_t$ and get $f_t$ and $g_t$ \> Full f.
4: Play $\mathbf{x}_t$ and get $f_t(\mathbf{x}_t)$ and $g_t(\mathbf{x}_t)$ \> Bandit f.
5: \> Primal RM update
6: Let $u^p_t : \mathbf{x} \mapsto v_t f_t(\mathbf{x}) - \langle \lambda_t, g_t(\mathbf{x}) \rangle$ \> Full f.
7: $u^p_t(\mathbf{x}_t) \leftarrow v_t f_t(\mathbf{x}_t) - \langle \lambda_t, g_t(\mathbf{x}_t) \rangle$ \> Bandit f.
8: $\mathcal{R}^p_.\text{OBSERVEUTILITY}(u^p_t)$ \> Full f.
9: \> Dual RM update
10: Let $u^d_t : \lambda \mapsto \langle \lambda, g_t(\mathbf{x}) \rangle$
11: $\mathcal{R}^d_.\text{OBSERVEUTILITY}(u^d_t(\mathbf{x}_t))$ \> Bandit f.
12: \> END
4 Analysis with stochastic constraints and adversarial rewards

We start by analyzing the performance of our meta-algorithm (Algorithm 1) when the reward and constraint functions are selected stochastically and adversarially, respectively.

Given $t \in [T]$ and $\eta \in (0, 1)$, we let $E_{t, \eta} := \sqrt{8t \log(18m^2/\eta)}$ be the value bounding differences between expectations and empirical means of constraint functions, obtained by applying the Azuma-Hoeffding inequality, and holding with probability at least $1 - \eta$. Given $\gamma \in (0, 1)$, we also let

$$M_\gamma := \frac{2}{\gamma} \sqrt{T} + \left(2 + \frac{3}{\gamma}\right) E_{t, \eta} + \left(1 + \frac{2}{\gamma}\right) E_{t, \eta}^\rho + \frac{1}{\gamma} E_{t, \eta}^\nu,$$

which is a recurring term related to the maximum violation that Algorithm 1 accepts in play phase.

First, we introduce a useful event $E$ that encompasses all the cases in which Algorithm 1 successfully terminates. Then, Lemma 4.2 shows that such an event holds with probability at least $1 - \delta$. In particular, $E$ holds when the regret bounds of $\mathcal{R}_1^\nu$ and $\mathcal{R}_2^\nu$ hold, and, additionally, the differences between expectations and empirical means of constraint functions are bounded as desired.

**Definition 4.1.** We denote with $E$ the event in which Algorithm 1 satisfies the following conditions (recall that $\eta = \delta/3$): (i) the regret incurred by $\mathcal{R}_1^\nu$ after $T_1$ rounds is upper bounded by $E_{T, \eta}$; (ii) the regret cumulated by $\mathcal{R}_2^\nu$ after the remaining $T - T_1$ rounds is upper bounded by $E_{T - T_1, \eta}$; and (iii) for every pair of rounds $t, t' \in [T]: t \leq t'$ and resource $i \in [m]$ it holds:

- $\sum_{t=t}^{t'} g_r,i(x_t) - \sum_{t=t}^{t'} \bar{g}_i(x_t) \leq E_{t', t, \eta}$,
- $\sum_{t=t}^{t'} \bar{g} \bar{g}_i(x_t) - \sum_{t=t}^{t'} \bar{g}_i(x_t) \leq E_{t', t, \eta}$,
- $\sum_{t=t}^{t'} g_r,i(x_t) - \sum_{t=t}^{t'} \bar{g}_i(x_t) \leq E_{t', t, \eta}$ for $\bar{g} \in \{\xi^*, \xi^\circ\}$.
- $\sum_{t=t}^{t'} \bar{g} \bar{g}_i(x_t) - \sum_{t=t}^{t'} \bar{g}_i(x_t) \leq E_{t', t, \eta}$.

**Lemma 4.2.** After running Algorithm 1 the event $E$ holds with probability at least $1 - \delta$.

Next, we lower bound the cumulative reward obtained by Algorithm 1 during the play phase. Intuitively, we show that, if the cumulative constraints violation is large, then the decisions $x_t$ in the first $T_1$ rounds provide a per-round reward much higher than that achievable by $\xi^*$. This allows us to employ the following recovery phase to decrease constraints violation cumulated in the play phase, while also ensuring that the cumulative regret stays low at the end of the algorithm. Formally:

**Lemma 4.3.** If event $E$ holds, then after round $T_1$ of Algorithm 1 the following inequality holds:

$$\sum_{t=1}^{T_1} f_i(x_t) \geq \sum_{t=1}^{T_1} f_i(\xi^*) + (T - T_1) - \frac{1}{p} E_{T_1, \eta} - \left(1 + \frac{2}{\gamma}\right) E_{T_1, \eta}^\nu + \frac{1}{p} E_{T_1, \eta}^\nu.$$

In the recovery phase, the only goal of Algorithm 1 is to decrease constraints violation. In the following Lemma 4.4, we show that, at each round of the recovery phase, the algorithm is “close” to satisfying (in expectation) all the constraints by at least $\rho$. Formally:

**Lemma 4.4.** If event $E$ holds, then after Algorithm 1 halts, the following holds for every $i \in [m]$:

$$\sum_{t=T_1+1}^{T} g_r,i(x_t) \leq -(T - T_1) \rho + 2 E_{T - T_1, \eta}^\nu + E_{T - T_1, \eta}^\nu.$$

Now, we are ready to present the two main results of this section. First, we provide a bound on the cumulative regret and constraints violation when the lower bound $\hat{\rho}$ is sufficiently large.

**Condition 4.5.** It holds that $\hat{\rho} \geq 2T^{-1/4}$.

Notice that, under Condition 4.5, $\hat{\rho} = \rho/2$. This gives us the following result:

**Theorem 4.6.** Suppose that functions $f_i$ and $g_i$ are selected adversarially and stochastically, respectively. If Condition 4.5 is satisfied, then, with probability at least $1 - \delta$, Algorithm 1 provides

$$RT \leq \frac{1}{\rho} E_{T, \eta} + \left(1 + \frac{2}{\gamma}\right) E_{T, \eta}^\nu + \frac{1}{p} E_{T}^\nu$$

and

$$V_T \leq M_{\hat{\rho}} + 2E_{T, \eta}^\nu + E_{\hat{\rho}} + E_{T, \eta}.$$

Finally, we also prove that even if Condition 4.5 is not satisfied, i.e., the lower bound $\hat{\rho}$ is not sufficiently large, the following holds:
As a first step to prove our result, we provide a lower bound on the cumulative reward of the primal
Appendix A for more details). Moreover, as we discuss later in Section 8, when restricted to the case
which both reward and constraint functions are selected stochastically.

\[ \sum_{t=1}^{T} f_t(x_t) \geq \frac{\rho}{1+\rho} \sum_{t=1}^{T} \min_{\bar{f}} E_{T,\eta}^{T} - \left( 1 + \frac{2}{\rho} \right) E_{T,\eta}^{T} + \frac{1}{\rho} E_{T}^{T} + V^{T} \leq M_T^{T} + 2E_{T,\eta}^{T} + E_{T,\eta}^{T} + E_{T}\eta. \]

Remark 4.8. Notice that, by using primal and dual RMs whose regret bounds are of the order of
\[ \tilde{O}(\sqrt{T}), \] Theorem 4.6 allows us to recover \[ O(\sqrt{T}/\hat{\rho}) \] regret and \[ O(\sqrt{T}/\hat{\rho}) \] constraints violation for
the case in which Condition 4.5 holds. Theorem 4.7 still provides \[ O(T^{3/4}) \] regret and constraints
violation when the condition is not met, which is necessary the case when \( \rho = 0 \).

5 Analysis with stochastic constraints and stochastic rewards

In this section, we focus on the case in which both reward and constraint functions are selected
stochastically. In this setting, we are able to show that Algorithm 1 never enters the recovery phase.
As we argue in Section 8, this is an important property for budget-management applications, since it
is related to the round in which the budget is fully depleted.

In order to prove our result, we extend the event \( E \) to capture also the Azuma-Hoeffding bounds for
the reward functions, which are stochastic in this setting. The core idea that we exploit to prove our
result is that we can think of the two RMs as if they are playing a stochastic repeated zero-sum game,
which is the repeated Lagrangian game whose functions are sampled according to the probability
measures \( \mu_{\bar{F}} \) and \( \mu_{G} \). By Theorem 2.3, strong duality holds, and the game has an equilibrium. Hence,
it is possible to show that the per-round utility of the primal RM is close to the value of the game,
which is \( \min_{\bar{F}} E^{T,\bar{g}} \). At the same time, it is possible to show that, if the cumulative constraints violation
becomes large during the play phase (and, thus, \( T_1 < T \)), then the per-round utility of the primal RM is
below \( \min_{\bar{F}} E^{T,\bar{g}} \), reaching a contradiction that proves the following theorem.

Theorem 5.1. Suppose that functions \( f_t \) and \( g_t \) are selected stochastically. With probability at least
\( 1 - \delta \), Algorithm 1 never enters the recovery phase, namely \( T_1 = T \).

Notice that regret bounds analogous to the ones in Theorems 4.6 and 4.7 also hold in the case in
which both reward and constraint functions are selected stochastically.

6 Analysis with adversarial constraints

In this section, we study settings in which the constraint functions \( g_t \) are selected adversarially. As
shown by Mannor et al. [36], it is impossible to obtain sublinear cumulative regret and constraints
violation when using our baseline, i.e., the best fixed strategy mixture \( \xi^{*} \) satisfying (in expectation)
the long-term constraints. However, we show that it is possible to achieve a \( \rho/(1 + \rho) \) fraction
of the cumulative reward obtained by always playing \( \xi^{*} \), while guaranteeing sublinear constraints
violation. The dependence of the approximation factor on the feasibility parameter \( \rho \) is similar to the
dependence on the per-round budget in problems with budget constraints (see the related works in
Appendix A for more details). Moreover, as we discuss later in Section 8, when restricted to the case
of budget constraints and adversarial reward/cost functions, our approximation factor matches the
state-of-the-art bounds provided by Castiglioni et al. [17].

As a first step to prove our result, we provide a lower bound on the cumulative reward of the primal
RM during the play phase. We show that it achieves at least a \( \rho/(1 + \rho) \) fraction of the value obtained
by the optimal solution in the first \( T_1 \) rounds. Moreover, the algorithm provides an additional utility
compensating for the last rounds in which the algorithm only focuses in satisfying the constraints.
Finally, we show that, in the recovery phase, the constraints are satisfied by at least \( \rho \) at each round,
up to a term related to the regret of \( R_{\xi}^{T} \) and \( R_{\xi}^{T} \), proving the following theorem.

Theorem 6.1. Suppose that functions \( f_t \) and \( g_t \) are selected adversarially. If Condition 4.5 is
satisfied, then, with probability at least \( 1 - \frac{2}{\rho} \), Algorithm 1 guarantees that the following holds:
\[
\sum_{t=1}^{T} f_t(x_t) \geq \frac{\rho}{1+\rho} \sum_{t=1}^{T} \min_{\bar{f}} E_{T,\eta}^{T} - \left( 1 + \frac{2}{\rho} \right) E_{T,\eta}^{T} + \frac{1}{\rho} E_{T}^{T} + V^{T} \leq M_T^{T} + 2E_{T,\eta}^{T} + E_{T}\eta.
\]

A similar result can be also derived for the case of stochastic rewards and adversarial constraints.
Corollary 6.2. Suppose functions \( f_t \) and \( g_t \) are selected stochastically and adversarially, respectively. If Condition 4.5 is satisfied, then, with probability at least \( 1 - \delta \), Algorithm 3 provides \( \sum_{t=1}^{T} f_t(x_t) \geq \frac{\rho}{1+\rho} \sum_{t=1}^{T} \text{OPT}_{f,g} - \left(1 + \delta \right) \sum_{t=1}^{T} \epsilon_{t}^{f} - \frac{1}{2} \sum_{t=1}^{T} \epsilon_{t}^{D} + \frac{1}{2} \sum_{t=1}^{T} \epsilon_{t}^{H} \) and \( \sum_{t=1}^{T} V_{t} \leq M_{\rho} + 2 \sum_{t=1}^{T} \epsilon_{t}^{D} + \epsilon_{t}^{H} + \epsilon_{t}^{H} \).

Remark 6.3. By using primal and dual RMs whose regret bounds are of the order of \( \tilde{O}(\sqrt{T}) \), Theorem 6.7 and Corollary 6.2 allows us to recover \( \sum_{t=1}^{T} f_t(x_t) \geq \frac{\rho}{1+\rho} \sum_{t=1}^{T} \text{OPT}_{f,g} - \tilde{O}(\sqrt{T}/\rho) \), and \( \tilde{O}(\sqrt{T}/\rho) \) constraints violation for the case in which Condition 4.5 holds.

7 How to get away with no knowledge about the feasibility parameter

We show how to extend Algorithm 1 in order to deal with settings in which a lower bound on the feasibility parameter \( \rho \) is not known, when functions \( g_t \) are selected stochastically. We propose Algorithm 3 which runs Algorithm 1 by first devoting a given number \( T_0 \) to inferring a suitable lower bound \( \hat{\rho} \) on the feasibility parameter \( \rho \). Ideally, we would like to have \( \hat{\rho} = \Omega(\rho) \), so as to recover bounds of the order \( \tilde{O}(\sqrt{T}/\rho) \). In particular, we show that we can run Algorithm 3 with \( T_0 = T^{1/2} \) to obtain an approximation of \( \rho \) that has an additive approximation error of the order \( T^{1/4} \).

This is sufficient to get \( \hat{\rho} = \Omega(\rho) \), since a good approximation of \( \rho \) is only needed when \( \rho \geq T^{1/4} \).

Let us remark that our approach only works when constraints functions \( g_t \) are selected stochastically. When these are chosen adversarially, it is easy to see that it is impossible to compute a lower bound on the feasibility parameter \( \rho \) by only using the first rounds. For instance, think of a setting in which \( \rho \) is very large in the first rounds, while it becomes small during the last ones.

To exploit the guarantees of Algorithm 1 presented in the previous sections, it is enough to show that, after the first \( T_0 \) rounds of Algorithm 3, \( \hat{\rho} \leq \rho \) holds with high probability.

Algorithm 3 Meta-Algorithm \((T, T_0, \delta)\)

1: \( \mathcal{R}^f \leftarrow \text{INIT}^{\delta}(\mathcal{X}, [-1, 1], \delta) \)
2: \( \mathcal{R}^D \leftarrow \text{INIT}^{\delta}(\Delta_{\text{opt}}, [-1, 1], 0) \)
3: \( t \leftarrow 1 \)
4: \( \text{while } t \leq T_0 \text{ do} \)
5: \( x_t \leftarrow \text{LAGRANGIAN GAME}(\mathcal{R}^f, \mathcal{R}^D, 0) \)
6: \( t \leftarrow t + 1 \)
7: \( \hat{\rho} \leftarrow -\frac{1}{T_0} \max_{t \in [m]} \sum_{i=1}^{T_0} g_t(x_t) + \epsilon_{T_0, \delta} \)
8: \( \text{Run Algorithm 1 with } T - T_0, 0, \hat{\rho} \)

Lemma 7.1. If \( T_0 = \sqrt{T} \), then Algorithm 3 guarantees that \( \hat{\rho} \leq \rho \) with probability at least \( 1 - \delta \).

In order to recover a good estimate of \( \rho \), we need the value of \( \rho \) to be sufficiently large.

Condition 7.2. It holds that \( \rho \geq \frac{2}{T_0} \left(2 \epsilon_{T_0, \delta} + 2 \epsilon_{T_0, \delta} + \epsilon_{T_0, \delta} \right) \).

Remark 7.3. Notice that, by using primal and dual RMs whose regret bounds are of the order \( \tilde{O}(\sqrt{T}) \) and by setting \( T_0 = \sqrt{T} \), Condition 7.2 is satisfied when \( \rho = \omega(T^{-1/4}) \).

Next, we show that \( \hat{\rho} = \Omega(\rho) \), which allows us to exploit the guarantees proved for Algorithm 1 in order to provide analogous ones for Algorithm 3. Formally:

Lemma 7.4. By setting \( T_0 = \sqrt{T} \), and assuming that Condition 7.2 is satisfied, after \( T_0 \) rounds of Algorithm 3 we have that \( \hat{\rho} \geq \rho/2 \) with probability at least \( 1 - 2\delta \).

By applying the results of the previous sections on the guarantees of Algorithm 1 and by using primal and dual RMs whose regret bounds are of the order \( \tilde{O}(\sqrt{T}) \), we get \( \tilde{O}(\sqrt{T}/\rho) \) regret and violation bounds, respectively, when the functions \( g_t \) are selected stochastically.

8 Applications to repeated auctions settings

Internet advertising platforms usually operationalize large auction markets by using proxy bidders that place bids in repeated auctions on the advertisers’ behalf. A proxy-bidder selects bids according to a budget-pacing mechanism, which manages the usage of the advertisers’ budget over time [1, 22, 9]. In this section, we discuss the application of our framework to budget-management in auctions, arguing that it can deal with more general constraints on ad slots allocation with respect to what is currently achievable with multiplicative pacing algorithms, which manage only knapsack constraints.
We consider the problem faced by a bidder who takes part in a sequence of repeated auctions. We focus on the case of *second-price* and *first-price* auctions, since they are the *de facto* standard in large Internet advertising platforms. At each round $t \in [T]$, the bidder observes their valuation $v_t$ from a finite set of $n_v$ possible valuations $\mathcal{V} \subset [0, 1]$. Such valuation models targeting preferences of the advertiser. Then, the bidder chooses a bid $b_t \in \mathcal{B}$, where $\mathcal{B} \subset [0, 1]$ is a finite set of $n_B$ possible bids such that $0 \in \mathcal{B}$ (i.e., the bidder is allowed to skip items without incurring in any cost). The utility of the bidder depends on the largest among competing bids, denoted by $\beta_i$. In particular, the utility is computed as $f_i(b_t) = (v_t - c_i(b_t)) \mathbb{1}\{b_t \geq \beta_i\}$, where the cost $c_i$ is such that $c_i(b_t) = \beta_i \mathbb{1}\{b_t \geq \beta_i\}$ in second-price auctions, and $c_i(b_t) = b_t \mathbb{1}\{b_t \geq \beta_i\}$ for first-price ones. Finally, the bidder has a target *per-round* budget of $\rho > 0$, which yields an overall budget $B := \rho T$ that limits the total spending over the $T$ rounds. In the case of budget-constrained bidding, a strictly feasible solution can be easily achieved by always bidding 0. Using the target per-round budget $\rho = B/T$ we can write the budget constraint as $\sum_{t \in [T]} g_t(b_t) \leq 0$, with $g_t(b) = c_t(b) - \rho$ for any $b \in \mathcal{B}$. Notice that, in this setting, we have the same feasibility parameter $\rho$ for both the stochastic and the adversarial case.

As a benchmark to evaluate the algorithm, we consider the best feasible static policy $\pi : \mathcal{V} \rightarrow \mathcal{B}$. The set of static policies can be represented by $\mathcal{X} := \mathcal{B}^{n_v}$, where a vector $b \in \mathcal{B}^{n_v}$ encodes the policy’s bids for each possible valuation. To apply our framework to this problem, it is sufficient to design a primal regret minimizer constructor (recall that, in order to design dual RMs, we can employ OMD).

We consider the problem faced by a bidder who takes part in a sequence of repeated auctions. We focus on the case of *second-price* and *first-price* auctions, since they are the *de facto* standard in large Internet advertising platforms. At each round $t \in [T]$, the bidder observes their valuation $v_t$ from a finite set of $n_v$ possible valuations $\mathcal{V} \subset [0, 1]$. Such valuation models targeting preferences of the advertiser. Then, the bidder chooses a bid $b_t \in \mathcal{B}$, where $\mathcal{B} \subset [0, 1]$ is a finite set of $n_B$ possible bids such that $0 \in \mathcal{B}$ (i.e., the bidder is allowed to skip items without incurring in any cost). The utility of the bidder depends on the largest among competing bids, denoted by $\beta_i$. In particular, the utility is computed as $f_i(b_t) = (v_t - c_i(b_t)) \mathbb{1}\{b_t \geq \beta_i\}$, where the cost $c_i$ is such that $c_i(b_t) = \beta_i \mathbb{1}\{b_t \geq \beta_i\}$ in second-price auctions, and $c_i(b_t) = b_t \mathbb{1}\{b_t \geq \beta_i\}$ for first-price ones. Finally, the bidder has a target *per-round* budget of $\rho > 0$, which yields an overall budget $B := \rho T$ that limits the total spending over the $T$ rounds. In the case of budget-constrained bidding, a strictly feasible solution can be easily achieved by always bidding 0. Using the target per-round budget $\rho = B/T$ we can write the budget constraint as $\sum_{t \in [T]} g_t(b_t) \leq 0$, with $g_t(b) = c_t(b) - \rho$ for any $b \in \mathcal{B}$. Notice that, in this setting, we have the same feasibility parameter $\rho$ for both the stochastic and the adversarial case.

As a benchmark to evaluate the algorithm, we consider the best feasible static policy $\pi : \mathcal{V} \rightarrow \mathcal{B}$. The set of static policies can be represented by $\mathcal{X} := \mathcal{B}^{n_v}$, where a vector $b \in \mathcal{B}^{n_v}$ encodes the policy’s bids for each possible valuation. To apply our framework to this problem, it is sufficient to design a primal regret minimizer constructor (recall that, in order to design dual RMs, we can employ OMD). This can be implemented by instantiating a regret minimizer EXP3.P [5] for each possible valuation in $\mathcal{V}$. Given a failure probability $\nu \in (0, 1)$, each RM guarantees a regret bound $O(\sqrt{T n_v \log(n_v/\nu)})$ with probability at least $1 - \nu$. Thus, given a desired failure probability $\eta \in (0, 1)$, by setting $\nu = \eta/n_v$, we get that, with probability at least $1 - \eta$, the bounds of all the RMs hold. Hence, by a union bound, we have that the regret of a primal RM is $E_{T, \eta} = O(n_v \sqrt{T n_v \log(n_v \eta^{-1})})$.

**Guaranteed budget completion in the stochastic case.** The crux of budget-pacing mechanisms is ensuring that the advertisers’ budget is not depleted too early (thereby missing potentially valuable future advertising opportunities), while being fully depleted within the planned duration of the campaign. Theorem 5.1 shows that, when inputs are generated according to some stochastic model, Algorithm 1 never enters the recovery phase. This is crucial in the context of budget-pacing mechanisms, because whenever the algorithm enters the recovery phase it will converge to always bid 0 in order to mitigate constraints violation. Therefore, the bidder could miss out on potentially valuable items. Moreover, if the platform wanted to guarantee that the bidder does not spend more than the budget $B$, it would be enough to set a *virtual budget* of $B - \tilde{O}(T^{1/2})$ to compensate for the potential constraints violation. Finally, we argue that, in large-scale markets, an individual bidder has almost no impact on prices, and, thus, stochastic behavior of costs is a reasonable assumption.

**Adversarial case.** Theorem 6.1 of [17] shows how to construct an algorithm that provides a $\rho$ fraction of the optimal utility for problems with budget constraints and adversarial inputs. The ratio $\rho/(1 + \rho)$ obtained in Theorem 6.1 matches such result. The latter assumes that rewards and costs are in $[0, 1]$, and, thus, $g_t \in [-\rho, 1 - \rho]$ (as they only model budget constraints). However, in our case we have $g_t \in [-1, 1]$. By normalizing the former range to match with ours, we get $g_t \in [-\rho/(1 - \rho), 1]$. Therefore, the feasibility parameter would be $\rho' = \rho/(1 - \rho)$. By rewriting our guarantees as a function of $\rho$, we get $\rho'/\rho = \rho'/(1 + \rho') = \rho$, which is the same guarantee of [17].

**Handling ROI constraints.** Traditional budget-pacing mechanisms (see, e.g., [11, 8]) are based on primal-dual algorithms that are near optimal in settings with knapsack constraints only, and they cannot be generalized to deal with other types of long-term constraints. However, there are many real-world situations in which guaranteeing other types of constraints is crucial for practical applications (see, e.g., [26, 25]). One example is the case of return on investment (ROI) constraints [6, 26, 32]. The recent work by Golrezaei et al. [25] presents a threshold-based algorithm for repeated second-price auctions under budget and ROI constraints. Our framework allows advertisers to reach a target ROI while keeping expenses under control also in the setting of repeated first-price auctions. In particular, given a target ROI $\omega$, we define the ROI constraints as $g_t(b_t) = (\omega - \nu_t/b_t) \mathbb{1}\{b_t \geq \beta_t\} \leq 0$. Then, it is enough to instantiate our framework as described before to immediately get that the cumulative violation of the budget and ROI constraints are upper bounded by $\tilde{O}(T^{1/2})$. This holds both in the fully stochastic and in the fully adversarial setting.

See Appendix C for a detailed discussion on the types of constraints that our framework can handle.
References


Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes] We present the goal of our paper and summarize the theoretical results. In the Original Contribution section, we underline the novelty of our work and in the Related Work section we compare our results with those in the literature (see also Table 1).
   (b) Did you describe the limitations of your work? [Yes] The assumptions characterizing the settings in which our results hold are clearly stated throughout the paper.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] The results presented in this work does not have any foreseeable negative societal consequence.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] In each theorem statement we provide all the assumptions needed for the result to hold.
   (b) Did you include complete proofs of all theoretical results? [Yes] They are provided in Appendix B and Appendix C.

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]