Regularized Online Allocation Problems: Fairness and Beyond

Santiago Balseiro [*] [1] [2] Haihao Lu [*] [3] [2] Vahab Mirrokni [2]

Abstract

Online allocation problems with resource constraints have a rich history in computer science and operations research. In this paper, we introduce the *regularized online allocation problem*, a variant that includes a non-linear regularizer acting on the total resource consumption. In this problem, requests repeatedly arrive over time and, for each request, a decision maker needs to take an action that generates a reward and consumes resources. The objective is to simultaneously maximize total rewards and the value of the regularizer subject to the resource constraints. Our primary motivation is the online allocation of internet advertisements wherein firms seek to maximize additive objectives such as the revenue or efficiency of the allocation. By introducing a regularizer, firms can account for the fairness of the allocation or, alternatively, punish under-delivery of advertisements—two common desiderata in internet advertising markets. We design an algorithm when arrivals are drawn independently from a distribution that is unknown to the decision maker. Our algorithm is simple, fast, and attains the optimal order of sub-linear regret compared to the optimal allocation with the benefit of hindsight. Numerical experiments confirm the effectiveness of the proposed algorithm and of the regularizers in an internet advertising application.

1. Introduction

Online allocation problems with resource constraints have abundant real-world applications and, as such, have been extensively studied in computer science and operations research. Prominent applications can be found in internet advertising and cloud computing, both of which are multi-billion dollar markets. In display advertising, for example, a publisher typically signs contracts with many advertisers agreeing to deliver a fixed number of impressions within a limited time horizon. Impressions arrive sequentially over time and the publisher needs to assign, in real time, each impression to one advertiser so as to maximize metrics such as the cumulative click-through rate or the number of conversions while satisfying contractual agreements on the number of impressions to be delivered (Feldman et al., 2010). In cloud computing, jobs arriving online need to be scheduled to one of many servers. Each job consumes resources from the server, which need to be shared with other jobs. The scheduler needs to assign jobs to servers to maximize metrics such as the cumulative revenue or efficiency of the allocation. When jobs' processing times are long compared to their arrival rates, this scheduling problem can be cast as an online allocation problem (Badanidiyuru et al., 2018).

The literature on online allocation problems focuses mostly on optimizing additively separable objectives such as the total click-throughout rate, revenue, or efficiency of the allocation. In many settings, however, decision makers are also concerned about ancillary objectives such as fairness across advertisers, avoiding under-delivery of impressions, balancing the load across servers, or avoiding saturating resources. These metrics are, unfortunately, non-separable and cannot be readily accommodated by existing algorithms that are tailored for additively separable objectives. Thus motivated, in this paper, we introduce the *regularized online allocation problem*, a variant that includes a non-linear regularized acting on the total resource consumption. The introduction of a regularizer allows the decision maker to simultaneously maximize an additively separable objective together with other metrics such as fairness and load balancing that are non-linear in nature.

More formally, we consider a finite horizon model in which requests arrive repeatedly over time. The decision maker is endowed with a fixed amount of resources that cannot be replenished. Each arriving requests is presented with a concave reward function and a consumption matrix. After observing the request, the decision makers needs to take an action that generates a reward and consumes resources. The objective of the decision maker is to maximize the sum of the cumulative reward and a regularizer that acts on the total resource consumption. (Our model can easily accommodate

[*] Equal contribution [1] Columbia University, New York, USA [2] Google Research, New York, USA [3] University of Chicago. Correspondence to: Santiago R. Balseiro <srb2155@columbia.edu>, Haihao Lu <haihao.lu@chicagobooth.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

a regularizer that acts on other metrics such as, say, the cumulative rewards by adding dummy resources.)

Motivated by practical applications, we consider an incomplete information model in which requests are drawn independently from a distribution that is *unknown to the decision maker*. That is, when a request arrives, the decision maker observes the reward function and consumption matrix of the request before taking an action, but does not get to observe the reward functions and consumption matrices of future requests until their arrival. For example, in display advertising, publishers can estimate, based on the attributes of the visiting user, the click-through rates of each advertiser before assigning an impression. However, the click-through rates of future impressions are not known in advance as these depend on the attributes of the unknown, future visitors. The objective of this paper is to design simple algorithms that attain low regret relative to the best allocation when all requests are known in advance.

1.1. Our Results

Duality theory has been successfully used to tackle online allocation problems with additively separable objectives because it allows to decouple a master, resource-constrained problem into simpler subproblems—one for each request. We show that similar techniques can be used to design algorithms for online allocation problems with a non-additively separable regularizer.

In particular, we propose a dual-descent algorithm that maintains a dual variable for each resource constraint. When a request arrives, the reward function is adjusted with the dual variables to capture the opportunity of consuming resources, and then actions are taken greedily with respect to the dual-variable-adjusted reward function. The dual variables are also used to determine an "ideal" resource consumption that optimizes the regularizer. A simple, yet key observation is that by comparing the actual resource expenditure of the current action to the ideal resource consumption from the regularizer, we can construct a noisy, unbiased subgradient of the dual objective function. Using these subgradients as inputs, our algorithm employs weighted online subgradient descent to update the dual variables after each request. We prove that the regret of our algorithm is of the order $O(T^{1/2})$, where T is number of time periods, when resources are scaled proportionally to the length of the horizon. This rate is unimprovable under our minimal assumptions on the input.

When updating the dual variables, it is required to project dual variables to the feasible set. In standard online allocation problems the dual feasible set is the non-negative orthant and the projection step is trivial. The introduction of a regularizer, however, alters the geometry dual feasible set. In many cases, the dual feasible set becomes more complex, which, in turn, results in a more difficult projection problem. By suitably picking the weights for subgradient descent, it is possible to adjust the update rule to better capture the geometry of the dual feasible set and obtain more tractable projection problems.

An advantage of our algorithm is that it is efficient and simple to implement. In many cases, the update rule can be implemented in linear time and there is no need to solve auxiliary convex optimization problems on historical data as in other methods in the literature.

To illustrate of our approach we discuss several regularizers that are useful in practice and numerically evaluate our algorithm on an internet advertising application using a max-min fairness regularizer. Our experiments confirm that our proposed algorithm attains $O(T^{1/2})$ regret as suggested by our theory, and showcase the trade-off between click-through rates and fairness: fairness can be significantly improved by reducing click-through rates by a small amount.

1.2. Related Work

Online allocation problems have been extensively studied in computer science and operations research literature. Table 1 summarizes the differences between our work and the existing literature on online allocation problems.

There is a stream of literature that studies online allocation problems under adversarial input models, i.e., when the incoming requests are adversarially chosen (Mehta et al., 2007; Feldman et al., 2009). We focus, instead, on stochastic models when the incoming requests are drawn i.i.d. from an unknown distribution.

Early work on online allocation with stochastic input models focus on linear reward functions, i.e., the case when the reward function is linear in the decision variable. Devanur and Hayes (2009) presented a two-phase dual training algorithm for linear reward function that is proportional to the resource consumption, which attains $O(T^{2/3})$ regret. Feldman et al. (2010) introduced a similar algorithm for more general linear reward functions which obtains the same order of regret. Later on, Agrawal et al. (2014) proposed a new dual-based algorithm that periodically solves a linear program using all data collected so far in order to update the dual variable, which improves the regret bound to $O(T^{1/2})$. Devanur et al. (2019) studied more complicated algorithms that not only obtain $O(T^{1/2})$ regret, but also yield near-optimal dependence on other parameters of the problems such as the number of resources. On the other hand, by a result of Arlotto and Gurvich (2019), $\Omega(T^{1/2})$ regret turns out to be lowest possible attainable regret under such settings. With additional assumptions on the input, the regret bound can be further improved. When the expected instance is non-degenerate, Jasin (2015) presented a new algorithm that attains $O(\log T)$ regret by periodically re-estimating

Papers	Objective	Resource constraints	Input model	Results
Mehta et al. (2007); Feldman et al. (2009)	Linear	Hard	Adversarial	Fixed comp. ratio
Devanur and Hayes (2009); Feldman et al. (2010); Agrawal et al. (2014); Devanur et al. (2019); Jasin (2015); Li and Ye (2019); Li et al. (2020)	Linear	Hard	Stochastic	Sublinear regret
Balseiro et al. (2020)	Nonlinear, separable	Hard	Stochastic	Sublinear regret
Jenatton et al. (2016); Agrawal and Devanur (2015)	Non-separable	Soft	Stochastic	Sublinear regret
Eghbali and Fazel (2016)	Non-separable	Soft	Adversarial	Fixed comp. ratio
Tan et al. (2020)	Non-separable	Hard	Adversarial	Fixed comp. ratio
This paper	Non-separable	Hard	Stochastic	Sublinear regret

Table 1: Comparison of our work with the existing literature on online allocation problems.

the distribution of requests. When the distribution of requests is absolutely continuous with uniformly bounded densities, Li and Ye (2019) presented a different algorithm that attains $O(\log T)$ regret. Their algorithm updates the dual variables by periodically solving a linear program using all data collected so far.

While the algorithms described above usually require solving large linear problem periodically, there is a recent line of work seeking simple algorithms that have no need of solving large linear program. Balseiro et al. (2020) studied a simple dual mirror descent algorithm for online allocation problems with concave reward functions, which attains $O(T^{1/2})$ regret. Our algorithm is similar to theirs in spirit, but our analysis is simpler as we do not need to explicitly bound the stopping time corresponding to the first time a resource is depleted. Their updates can be computed in linear time, and avoids the need of solving large linear program. Soon afterward, Li et al. (2020) presented a similar fast algorithm that attains $O(T^{1/2})$ regret for linear reward. Our proposed algorithm falls into this category: the update per iteration can be efficiently computed in linear time and there is no need to solve large convex optimization problems. A major difference is that we allow for a regularizer in the objective, which provides us with greater modeling power while increasing the analysis difficulty.

While most of the literature on online allocation focuses on maximizing an additively separable objective, other features of the allocation, such as fairness and load balancing, sometimes are crucial to the decision maker. For example, fairness is a central concept in welfare economics. Different reasonable metrics of equity have been proposed and studied: max-min fairness, which maximizes the reward of the worst-off agents (Nash Jr, 1950; Bansal and Sviridenko, 2006); proportional fairness, which makes sure that there is no alternative allocation that can lead to a positive aggregate proportional change for each agent (Azar et al., 2010; Bateni et al., 2018); or α-fairness, which generalizes the previous notions (Mo and Walrand, 2000; Bertsimas et al., 2011; 2012), and allows to recovers max-min fairness and proportional fairness as special cases when $\alpha = \infty$ and $\alpha = 1$, respectively. The above line of work focuses on optimizing the fairness of an allocation problem in either static settings, adversarial settings, or stochastic settings that are different to ours. In contrast, our framework is concerned with maximizing an additively separable objective but with an additional regularizer corresponding to fairness (or other desired ancillary objective).

The regularizer can be viewed as a reward that is not separable over time. The existence of such non-separable regularizer makes the theoretical analysis much harder than the one without the regularizer. For example, compared to Balseiro et al. (2020), one fundamental difficulty in the analysis is that the dual variables, which correspond to the opportunity cost that a unit of resource is consumed, can be negative due to the existence of the regularizer. This precludes the stopping time analysis introduced in Balseiro et al. (2020) from applying in our result. We come up with a new analysis that overcomes that difficulty by utilizing the complementary slackness condition to control the performance up to the stopping time.

Indeed, there have been previous works studying online allocation problems with non-separable rewards, but most of them are under very restrictive assumptions, precluding most of the interesting examples we present in Section 2.1. Jenatton et al. (2016) studies a problem with soft resource constraints in which violations of the constraints are penalized using a non-separable regularizer. Thus, different to our paper, they allow the resource constraints to be violated. They provide a similar algorithm to ours, but their regret bounds depend on the optimal solution in hindsight, which, in general, it is hard to control. Eghbali and Fazel (2016) consider maximizing a non-separable concave reward and requires the reward to be monotonic. They do not consider hard resource constraints and, moreover, they consider an adversarial input model. They provide similar primal-dual algorithms that are shown to attain fixed competitive ratios. Tan et al. (2020) studies a similar problem as our regularized online allocation setting, but restrict attention to one resource. Compared to the above works, we consider stochastic inputs, which allow us to attain sublinear regret, while they consider adversarial inputs under which sublinear regret is not attainable.

Another related work to ours is Agrawal and Devanur (2015), where the focus is to solve general online stochastic convex program that allows general concave objectives and convex constraints. When the value of the benchmark is known, they present fast algorithms; otherwise, their algorithms require periodically solving large convex optimization problems with the data collected so far to obtain a good estimate of the benchmark. In principle, our regularized online allocation problem (see Section 2 for details) can be reformulated as an instance of the online stochastic convex program presented in Agrawal and Devanur (2015). Such reformulation makes the algorithms proposed in Agrawal and Devanur (2015) more complex than ours as they require keeping track of additional dual variables and solving convex optimization program on historical data (unless the optimal value of the objective is known). Moreover, their algorithm treat resources constraints as soft, i.e., they allow constraints to be violated and then prove that constrains are violated, in expectation, by an amount sublinear in the number of time periods T. Instead, in our setting, resource constraints are hard and cannot be violated, which is a fundamental requirement in many applications. Additionally, our proposed algorithm is simple, fast, and does not require estimates of the value of the benchmark nor solving large convex optimization problems.

2. Problem Formulation

We consider the following generic convex problem with a finite horizon of T time periods, resource constraints, and a regularizer r on the resource consumption:

$$\begin{aligned}(O):\quad \max_{x:x_t\in\mathcal{X}} & \sum_{t=1}^{T} f_t(x_t) + Tr\left(\frac{1}{T}\sum_{t=1}^{T} b_t x_t\right) \\ \text{s.t.} & \sum_{t=1}^{T} b_t x_t \le T\rho\ ,\end{aligned} \tag{1}$$

where $x_t \in \mathcal{X} \subseteq \mathbb{R}^d$ is the action at time t, $f_t \in \mathbb{R}^d \to \mathbb{R}_+$ is the non-negative concave reward function received at time t, $b_t \in \mathbb{R}_+^{m\times d}$ is the entry-wise non-negative cost matrix received at time t, $\rho \in \mathbb{R}_{++}^m$ is the positive resource constraint vector, r is a concave regularizer on the consumption. The assumption $b_t \ge 0$ implies that we cannot replenish resources once they are consumed. We assume that the action set $\mathcal{X}$ is a convex and compact set in $\mathbb{R}_+^d$, and $0 \in \mathcal{X}$. The above assumption implies that we can only take non-negative actions. Moreover, we can always take a void action by choosing $x_t = 0$ in order to make sure we do not exceed the resource constraints. This guarantees the existence of a feasible solution.

We assume the request (f_t, b_t) at time t is generated i.i.d. from an unknown distribution $\mathcal{P} \in \Delta(\mathcal{S})$ with finite support $\mathcal{S} = \{(f^1, b^1), \ldots, (f^n, b^n)\}$ and where $\Delta(\mathcal{S})$ is the space of all probability distributions over $\mathcal{S}$.

In the online setting, at each time period $1 \le t \le T$, we receive a request (f_t, b_t), and we use an algorithm A to make a real-time decision x_t based on the current request (f_t, b_t) and the previous history $\mathcal{H}_{t-1} := \{f_s, b_s, x_s\}_{s=1}^{t-1}$, i.e.,

$$x_t = A(f_t, b_t|\mathcal{H}_{t-1})\ . \tag{2}$$

Moreover, algorithm A must satisfy constraints $\sum_{s=1}^{t} b_s x_s \le \rho T$ and $x_t \in \mathcal{X}$ for every $t \le T$. In particular, we define the expected reward of an algorithm A over distribution $\mathcal{P} \in \Delta(\mathcal{S})$ as

$$R(A|\mathcal{P}) = \mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{T} f_t(x_t) + Tr\left(\frac{1}{T}\sum_{t=1}^{T} b_t x_t\right)\right]\ ,$$

where x_t is computed by (2). The baseline we compare with is the expected reward of the optimal solution when all requests are known in advance, which is also referred as the offline problem in the computer science literature. This amounts to solving for the optimal allocation under full information of all requests and then taking expectations over all possible realizations:

$$\mathrm{OPT}(\mathcal{P}) = \mathbb{E}_{\mathcal{P}}\left[\begin{array}{ll}\max\limits_{x_t\in\mathcal{X}} & \sum\limits_{t=1}^{T} f_t(x_t) + Tr\left(\frac{1}{T}\sum\limits_{t=1}^{T} b_t x_t\right) \\ \text{s.t.} & \sum\limits_{t=1}^{T} b_t x_t \le T\rho\end{array}\right] \tag{3}$$

Our goal is to a design an algorithm A that attains low regret while satisfying above constraints. We measure the regret of an algorithm as the worst-case difference, over distributions in $\Delta(\mathcal{S})$, between the expected performance of the benchmark and the algorithm:

$$\mathrm{Regret}(A) = \sup_{\mathcal{P} \in \Delta(\mathcal{S})} \left\{ \mathrm{OPT}(\mathcal{P}) - R(A|\mathcal{P}) \right\} .$$

2.1. Examples of the Regularizer

We now describe some examples of the regularizer. First, by setting the regularizer to zero, we recover a standard online allocation problem.

Example 1. ***(No regularizer)*** *When the regularizer is $r(a) = 0$, we recover the non-regularized problem.*

Our next example allows for max-min fairness guarantees, which have been studied extensively in the literature (Nash Jr, 1950; Bansal and Sviridenko, 2006; Mo and Walrand, 2000; Bertsimas et al., 2011; 2012). Here we state the regularizer in terms of consumption. In many settings, however, it is reasonable to state the fairness regularizer in terms of other quantities such as the cumulative utility of advertisers. As discussed in Example 6, such regularizers can be easily accommodated by introducing dummy resource constraints.

Example 2. ***(Max-min Fairness)*** *The regularizer is defined as $r(a) = \lambda \min_j (a_j/\rho_j)$, i.e., the minimum relative consumption. This regularizer imposes fairness on the consumption between different advertisers, making sure that no advertiser gets allocated a too-small number of ad slots. Here $\lambda > 0$ is parameter that captures that importance of the regularizer relative to the rewards.*

In applications like cloud computing, load should be balanced across resources to avoid congestion. The following regularizer is reminiscent of the makespan objective in machine scheduling.

Example 3. ***(Load Balancing)*** *The regularizer is defined as $r(a) = -\lambda \max_j (a_j/\rho_j)$, i.e., the negative maximum relative consumption. This regularizer guarantees that consumption is evenly distributed across resources by making sure that no resource is too demanded.*

In some settings, the cost of utilizing resources is non-linear and convex because of decreasing returns to scale. The next regularizer allows to capture situations in which the cost of utilizing resources increases as they become saturated.

Example 4. ***(Hinge Loss of Consumption)*** *The regularizer is defined as $r(a) = -\sum_{j=1}^m c_j \max(a_j - t_j, 0)$, a hinge loss function with thresholds $t_j \in [0, \rho_j]$ and penalties c_j. This regularizer can be used when there is an extra variable cost c_j for each unit of resource consumed above a threshold t_j.*

To maximize reach, internet advertisers, in some cases, prefer that their budgets are spent as much as possible or their reservation contracts are delivered as many impressions as possible. We can incorporate these features by having the firm pay a goodwill penalty whenever targets are not met.

Example 5. ***(Mirrored Hinge Loss of Consumption)*** *The regularizer is defined as $r(a) = -\sum_{j=1}^m c_j \max(t_j - a_j, 0)$, a mirrored hinge loss function with thresholds $t_j \in [0, \rho_j]$ and penalties c_j. This regularizer can be used when the advertiser j would like to spend at least t_j and the firm pays a penalty c_j for under-delivering.*

The regularizers in the previous examples act exclusively on resource consumption. The next example shows that by adding dummy resource constraints that never bind, it is possible to incorporate regularizers that act on other quantities.

Example 6. ***(Santa Claus Regularizer from Bansal and Sviridenko 2006)*** *The Santa Claus regularizer intends to make sure the minimal reward of each advertiser is not too small. Here we consider reward function $f_t(x_t) = \sum_{j=1}^m (q_t x_t)_j$, where $q_t \in \mathbb{R}^{m \times d}$, and $q_t x_t \in \mathbb{R}^m$ is the reward vector for the m advertiser with decision x_t. While the reward function f_t measures the total reward for all advertisers, the regularizer intends to make sure the minimal reward of the advertisers is not too small. To reduce such problem to our setting, we first add auxiliary budget constraints $\sum_{t=1}^T q_t x_t \le T \bar{f} e$, and then regularizer is $r\left(\frac{1}{T}\sum_{t=1}^T q_t x_t, \frac{1}{T}\sum_{t=1}^T b_t x_t\right) = \frac{1}{T}\min_j \sum_{t=1}^T (q_t x_t)_j$, where $\bar{f}$ is an upper bound of the possible reward.*

2.2. The Dual Problem

Our algorithm is of dual-descent nature and, thus, the Lagrangian dual problem of (3) and its constraint set play a key role. We construct a Lagragian dual of (3) in which we move the constraints to the objective using a vector of Lagrange multipliers $\mu \in \mathbb{R}^m$. For $c \in \mathbb{R}^d$ and $\mu \in \mathbb{R}^m$ we define

$$\begin{aligned} f^*(c) &:= \sup_{x \in \mathcal{X}} \{ f(x) - c^\top x \} , \text{ and} \\ r^*(-\mu) &:= \sup_{a \le \rho} \{ r(a) + \mu^\top a \} , \end{aligned} \tag{4}$$

as the conjugate function of $f(x)$ restricted to $\mathcal{X}$[1] and the conjugate function of $r(a)$ restricted to $\{a | a \le \rho\}$, respectively. Define $\mathcal{D} = \{\mu \in \mathbb{R}^m \mid r^*(-\mu) < +\infty\}$ as the

[1] More precisely, $f^*(-c)$ is the conjugate function of $-f(x) + \mathbf{1}\{x \in \mathcal{X}\}$ with the standard definition of conjugate functions, where $\mathbf{1}\{x \in \mathcal{X}\}$ is the indicator function of the constraint. We redefine the conjugate function for simplicity.

set of dual variables for which the conjugate of the regularized is bounded. For a given distribution $\mathcal{P}$, define the Lagrangian dual function $D(\mu|\mathcal{P}) : \mathcal{D} \to \mathbb{R}$ as

$$D(\mu|\mathcal{P}) := \mathbb{E}_{(f,b)\sim\mathcal{P}}\left[f^*(b^\top \mu)\right] + r^*(-\mu) \ ,$$

then the following result shows that $D(\mu|\mathcal{P})$ provides a valid upper bound to OPT($\mathcal{P}$). All missing proofs are available in the appendix.

Proposition 1. *It holds for any $\mu \in \mathcal{D}$ that OPT($\mathcal{P}$) $\le$ $TD(\mu|\mathcal{P})$.*

Furthermore, we call

$$(D): \ \inf_{\mu\in\mathcal{D}} TD(\mu|\mathcal{P}) \ , \tag{5}$$

the dual problem to (1). As mentioned before, the feasible region of the dual problem, as given by $\mathcal{D}$, together with the conjugate of the regularizer plays a key role in our algorithm and in our regret analysis. The following result provides some useful properties of the set $\mathcal{D}$.

Lemma 1. *The set $\mathcal{D}$ is convex and positive orthant is inside the recession cone of $\mathcal{D}$, i.e., $\mathbb{R}^d_+ \subseteq \mathrm{recc}(\mathcal{D})$.*

Proof. Convexity follows from Proposition 1.1.6 of Bertsekas (2009). We prove the second part. Suppose $\mu \in \mathcal{D}$, namely $\max_{a\le\rho}\{r(a) + \mu^\top a\} < +\infty$. Then it holds for any $e \in \mathbb{R}^d_+$ and $\lambda > 0$ that

$$\begin{aligned}\max_{a\le\rho}\{r(a) + (\mu+\lambda e)^\top a\} &\le \max_{a\le\rho}\{r(a) + \mu^\top a\} + \lambda e^\top \rho \\ &< +\infty \ ,\end{aligned}$$

thus $\mu + \lambda e \in \mathcal{D}$, which finishes the proof by definition of recession cone. □

We next give explicit characterizations of the constraint set $\mathcal{D}$ and the conjugate r^* for the sample regularizers stated in Section 2.1.

2.3. The Constraint Set $\mathcal{D}$ for the Examples

The next proposition presents the conjugate functions r^*, the corresponding domain $\mathcal{D}$, and optimal actions $a^*(-\mu) \in \arg\max_{a\le\rho}\{r(a) + \mu^\top a\}$ for the first five examples stated in Section 2.1.

Proposition 2. *The following hold:*

- ***Example 1**: If $r(a) = 0$, then $\mathcal{D} = \mathbb{R}^m_+$ and, for $\mu \in \mathcal{D}$, $r^*(-\mu) = \mu^\top \rho$ and $a^*(-\mu) = \rho$.*
- ***Example 2**: If $r(a) = \lambda \min_j(a_j/\rho_j)$, then $\mathcal{D} = \left\{\mu \in \mathbb{R}^m \mid \sum_{j\in S}\rho_j\mu_j \ge -\lambda \ \forall S \subseteq [m]\right\}$, and, for $\mu \in \mathcal{D}$, $r^*(-\mu) = \rho^\top\mu + \lambda$ and $a^*(-\mu) = \rho$.*
- ***Example 3**: If $r(a) = -\lambda \max_j(a_j/\rho_j)$, then $\mathcal{D} = \left\{\mu \ge 0 \mid \sum_{j=1}^m \rho_j\mu_j \ge \lambda\right\}$, and, for $\mu \in \mathcal{D}$, $r^*(-\mu) = \rho^\top\mu + \lambda$ and $a^*(-\mu) = \rho$.*
- ***Example 4**: If $r(a) = -\sum_{j=1}^m c_j \max(a_j - t_j, 0)$, then $\mathcal{D} = \mathbb{R}^m_+$ and, for $\mu \in \mathcal{D}$, $r^*(-\mu) = \mu^\top t + \sum_{j=1}^m(\rho_j - t_j)\max(\mu_j - c_j, 0)$ and $a^*_j(-\mu) = t_j$ if $\mu_j \in [-c_j, 0)$ and $a^*_j(-\mu) = \rho_j$ if $\mu_j \ge 0$.*
- ***Example 5**: If $r(a) = -\sum_{j=1}^m c_j \max(t_j - a_j, 0)$, then $\mathcal{D} = \left\{\mu \in \mathbb{R}^M \mid \mu \ge -c\right\}$ and, for $\mu \in \mathcal{D}$, $r^*(-\mu) = \mu^\top t + \sum_{j=1}^m(\rho_j - t_j)\max(\mu_j, 0)$ and $a^*_j(-\mu) = t_j$ if $\mu_j \in [-c_j, 0)$ and $a^*_j(-\mu) = \rho_j$ if $\mu_j \ge 0$.*

Algorithm 1: Dual Subgradient Descent Algorithm

```
Input: Initial dual solution $\mu_0$, total number of time periods $T$, initial resources $B_0 = T\rho$, weight vector $w \in \mathbb{R}^m_{++}$, and step-size $\eta$.
for $t = 0, \dots, T-1$ do
    Receive $(f_t, b_t) \sim \mathcal{P}$.
    Make the primal decision and update the remaining resources:
        $\tilde{x}_t = \arg\max_{x\in\mathcal{X}}\{f_t(x) - \mu_t^\top b_t x\}$ ,
        $x_t = \begin{cases} \tilde{x}_t & \text{if } b_t\tilde{x}_t \le B_t \\ 0 & \text{otherwise} \end{cases}$ ,
        $a_t = \arg\max_{a\le\rho}\{r(a) + \mu_t^\top a\}$
        $B_{t+1} = B_t - b_t x_t$.
    Obtain a stochastic subgradient of $D(\mu_t|\mathcal{P})$:
        $\tilde{g}_t = -b_t\tilde{x}_t + a_t$ .
    Update the dual variable by weighted, projected subgradient descent:
        $\mu_{t+1} = \arg\min_{\mu\in\mathcal{D}}\langle\tilde{g}_t, \mu\rangle + \frac{1}{2\eta}\|\mu - \mu_t\|_w^2$ .   (6)
end
```

3. Algorithm

Algorithm 1 presents the main algorithm we study in this paper. Our algorithm keeps a dual variable $\mu_t \in \mathbb{R}^m$ for each resource that is updated using subgradient descent, which is the workhorse algorithm of online convex optimization (Hazan et al., 2016).

At time t, the algorithm receives a request (f_t, b_t), and computes the optimal response $\tilde{x}_t$ that maximizes an opportunity cost-adjusted reward of this request based on the current dual solution μ_t. It then takes this action (i.e., $x_t = \tilde{x}_t$) if the action does not exceed the resource constraint, otherwise it takes a void action (i.e., $x_t = 0$). Additionally, it chooses a target resource consumption a_t by maximizing the opportunity-cost adjusted regularized (in Proposition 2 we give closed-form solutions for a_t for the regularizers we consider). Notice that it follows from the definition of conjugate function (4) that $-b_t\tilde{x}_t \in \partial f_t^*(b_t^\top \mu_t)$ and $a_t \in \partial r^*(-\mu_t)$. Thus $\tilde{g}_t := -b_t\tilde{x}_t + a_t$ is an unbiased stochastic estimator of a subgradient of the dual problem $D(\mu|\mathcal{P})$ at μ_t:

$$\begin{aligned}\mathbb{E}_{\mathcal{P}}\left[\tilde{g}_t\right] &= \mathbb{E}_{\mathcal{P}}\left[-b_t\tilde{x}_t + a_t\right] \\ &\in \mathbb{E}_{(f,b)\sim\mathcal{P}}\left[\partial f^*(b^\top \mu_t)\right] + r^*(-\mu_t) \in \partial D(\mu_t|\mathcal{P}) \,.\end{aligned}$$

Finally, the algorithm utilizes $\tilde{g}_t$ to update the dual variable by performing an online subgradient descent step (6) with step-size η and weight w. The descent step (6) can be interpreted as minimizing over $\mathcal{D}$ a first-order Taylor expansion of the dual objective plus a term that penalizes movement from the incumbent solution μ_t using the weighted ℓ_2-norm $\|\cdot\|_w$, which is given by $\|x\|_w^2 = \sum_{j=1}^m w_j x_j^2$ for some weight vector $w \in \mathbb{R}_{++}^m$.

Algorithm 1 only takes an initial dual variable and a step-size as inputs, and is thus simple to implement. In some cases, though the constraint $\mathcal{D}$ can be complicated, a proper choice of the weight w may make the descent step (6) easily computable (in linear time or with a closed-form solution). We discuss some good choices for the weight w for Examples 2-5 in Appendix D. In particular, in Example 2, the dual feasible set $\mathcal{D}$ has an exponential number of constraints, but using weights $w_j = \rho_j^2$ we can cast (6) as quadratic program with a linear number of constraints. In Example 3, the constraint $\mathcal{D}$ is a simple polytope, and we can again cast (6) as quadratic program with a linear number of constraints. Finally, in examples 4 and 5, the set $\mathcal{D}$ is a simple box constraint, and we can recover projected gradient descent by using the un-weighted Euclidean norm.

4. Regret Bound

In this section, we present the worst-case regret bound of Algorithm 1 for solving (1). First we state the assumptions required in our analysis.

Assumption 1. *(Assumptions on the support of the distribution). There exists $\bar{f} \in \mathbb{R}_+$ and $\bar{b} \in \mathbb{R}_+$ such that for all requests $(f, b) \in \mathcal{S}$ in the support, it holds $f(x) \le \bar{f}$ for all $x \in \mathcal{X}$ and $\|bx\|_{w,*} \le \bar{b}$ for all $x \in \mathcal{X}$.*

The upper bounds $\bar{f}$ and $\bar{b}$ impose regularity on the space of requests, and will appear in the regret bound. In the assumption above, we denote by $\|\cdot\|_{w,*}$ the dual norm of $\|\cdot\|_w$, which is given by $\|x\|_{w,*}^2 = \sum_{j=1}^m x_j^2/w_j$.

Assumption 2. *(Assumptions on the regularizer r). We assume the regularizer r and the set $\mathcal{D} = \{\mu \in \mathbb{R}^m \mid r^*(-\mu) < +\infty\}$ satisfy:*

1. *The function $r(a)$ is concave.*
2. *The function $r(a)$ is L-Lipschitz continuous in the $\|\cdot\|_{w,*}$-norm on its effective domain, i.e., $|r(a_1) - r(a_2)| \le L\|a_1 - a_2\|_{w,*}$ for any $a_1, a_2 \le \rho$.*
3. *There exists a constant $\bar{a}$ such that for any $\mu \in \mathcal{D}$, there exists $a \in \arg\max_{a \le \rho}\{r(a) + \mu^\top a\}$ such that $\|a\|_{w,*} \le \bar{a}$.*
4. *There exist $\bar{r}$ and $\underline{r}$ such that for any $x \in \mathcal{X}$ and $(b, f) \in \mathcal{S}(\mathcal{P})$ that $\underline{r} \le r(bx) \le \bar{r}$.*

The first part of assumption is required to guarantee that the optimization problem is convex. The third part guarantees that the conjugate of the regularizer has bounded subgradient, while the last part imposes that the regularizer is bounded. Proposition 2 in Appendix 2.3 immediately implies that the assumption is satisfied by all examples we consider in this paper.

The previous assumptions imply, among other things, that the projection step (6) of the algorithm always admits a solution. First, continuity of the regularizer implies that $\mathcal{D}$ is closed by Proposition 1.1.6 of Bertsekas (2009). The objective is continuous and coercive. Therefore, the projection problem admits an optimal solution by Weierstrass theorem.

The next theorem presents the worst-case regret bound of Algorithm 1.

Theorem 1. *Consider Algorithm 1 with step-size $\eta \ge 0$ and initial solution $\mu_0 \in \mathcal{D}$. Suppose Assumptions 1-2 are satisfied. Then, it holds for any $T \ge 1$ that*

$$\text{Regret}(A) \le C_1 + C_2\eta T + \frac{C_3}{\eta} \,. \tag{7}$$

where $C_1 = (\bar{f} + \bar{r} + L(\bar{b} + \bar{a}) - \underline{r})/\underline{\rho}$, $C_2 = (\bar{b} + \bar{a})^2/2$, $C_3 = (L + C_1\|w\|_\infty^{1/2})^2 + \|\mu_0\|_w^2$, and $\underline{\rho} = \min_{j\in[m]} \rho_j$.

While the result has a similar flavor to Balseiro et al. (2020), we would like to highlight that their stopping-time analysis does not work for our setting with the regularizer because the dual variables can be strictly negative with the existence of a regularizer.

Some observations are in order. First, the previous result implies that, by choosing a step-size of order $\eta \sim T^{-1/2}$, Algorithm 1 attains regret of order $O(T^{-1/2})$ when the length of the horizon and the initial amount of resources are simultaneously scaled. Second, Lemma 1 from Arlotto and Gurvich (2019) implies that one cannot hope to attain regret lower than $\Omega(T^{-1/2})$ under our assumptions. (Their result

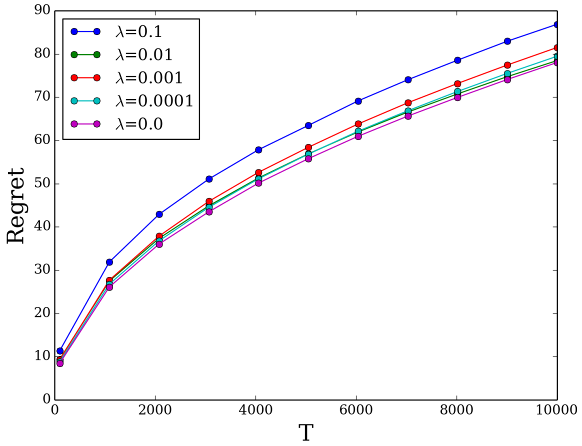

Figure 1: Plot of the regret versus the horizon T for the regularized online allocation problem (8) with different regularization levels.

holds for the case of no regularizer, which is a special case of our setting.) Therefore, Algorithm 1 attains the optimal order of regret.

We prove Theorem 1 in three steps. Let τ_A be the stopping time corresponding to the first time that a resource is depleted. The first step involves lower bounding the cumulative reward of the algorithm up to τ_A in terms of the dual objective (evaluated at the average dual variable used by the algorithm) minus a complementary slackness term. Until the stopping time τ_A, the algorithm performs standard subgradient descent steps on the dual function and, as a consequence, seeks to minimize complementary slackness. In the second step, we upper bound the complementary slackness term using standard results for online subgradient descent. By picking a suitable point in the dual space and using the theory of convex conjugates, we can relate the bound on the complementary slackness term to the value of regularizer. In the last step, we put everything together by using Proposition 1 to upper bound the optimal performance in terms of the dual function and then controlling the stopping time τ_A.

5. Numerical Experiments

In this section, we present numerical experiments on a display advertisement allocation application regularized by max-min fairness on consumption (Example 2).

Dataset. We utilize the display advertisement dataset introduced in Balseiro et al. (2014). They consider a publisher who has agreed to deliver ad slots (the requests) to different advertisers (the resources) so as to maximize the cumulative click-through rates (the reward) of the assignment. In their paper, they estimate click-through rates using mixtures of log-normal distributions. We adopt their parametric model as a generative model and sample requests from their estimated distributions. We consider publisher 2 from their dataset, which has $m = 12$ advertisers. Furthermore, in our experiments, we rescale the budget ρ so that $\sum_{j=1}^{m} \rho_j = 1.5$ in order to make sure that the max-min fairness is strictly less than 1.

Regularized Online Problem. The goal here is to design an online allocation algorithm that maximizes the total expected click-through rate with a max-min fairness regularizer on resource consumption as in Example 2. Advertiser $j \in [m]$ can be assigned at most $T\rho_j$ ad slots and the decision variables lie in the simplex $\mathcal{X} = \{x \in \mathbb{R}_+^m : \sum_{j=1}^{m} x_j \le 1\}$. Denoting by $q_t \in \mathbb{R}^m$ the click-through rate of the t-th ad slot T, we have that the benchmark is given by:

$$\begin{aligned} \max_{x:x_t\in\mathcal{X}} \sum_{t=1}^{T} q_t^\top x_t + \lambda \min_{j=1,\dots,m} \left(\sum_{t=1}^{T} (x_t)_j / \rho_j \right) \\ \text{s.t.} \sum_{t=1}^{T} x_t \le T\rho \,, \end{aligned} \tag{8}$$

where λ is the weight of the regularizer. In the experiments, we consider the regularization levels $\lambda \in \{0, 0.1, 0.01, 0.001, 0.0001\}$ and lengths of horizon $T \in \{10^2, 10^3, 2 \cdot 10^3, \dots, 10^4\}$.

Implementation Details. In the numerical experiments, we implemented Algorithm 1 with weights $w_j = \rho_j^2$ and step-size $0.01 \cdot T^{-1/2}$. The dual update (6) is computed by solving a convex quadratic program as stated in Appendix D using *cvxpy* (Diamond and Boyd, 2016). For each regularization level λ and time horizon T, we randomly choose T samples from their dataset that are fed to Algorithm 1 sequentially. In order to compute the regret, we utilize the dual objective evaluated at the average dual $D(\frac{1}{T}\sum_{t=1}^{T} \mu_t)$ as an upper bound to the benchmark. We report the average cumulative reward, the average max-min consumption fairness, and the average regret of 100 independent trials in Figure 1 and Figure 2. Ellipsoids in Figure 2 (b) give 95% confidence regions for the point estimates.

Observations. Consistent with Theorem 1, Figure 1 suggests that regret grows at rate $O(\sqrt{T})$ for all regularization levels. Figure 2 presents the trade-off between reward and fairness with the 95% confidence ellipsoid. In particular, we can double the max-min fairness while sacrificing only about 4% of the reward by choosing $\lambda = 0.01$. This showcases that fairness can be significantly improved by solving the regularized problem with a small amount of reward reduction.

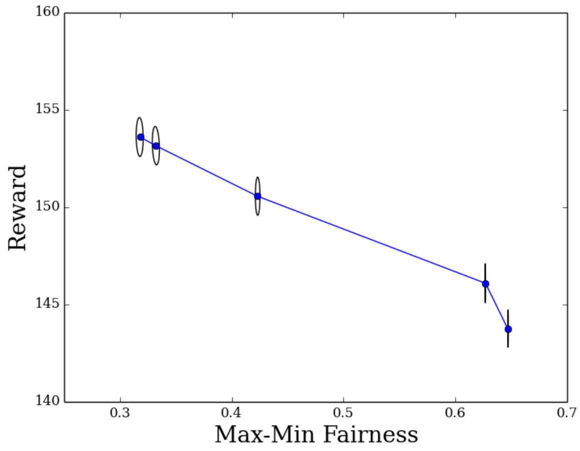

Figure 2: Plot of the reward $\sum_{t=1}^{T} q_t x_t$ versus the max-min fairness $\min_{j=1,\ldots,m}\left(\sum_{t=1}^{T}(x_t)_j/T\rho_j\right)$. Dots from left to right corresponds to regularization levels $\lambda = 0.0, 0.0001, 0.001, 0.01, 0.1$, respectively.

6. Conclusion and Future Directions

In this paper, we introduce the *regularized online allocation problem*, a novel variant of the online allocation problem that allows for regularization on resource consumption. We present multiple examples to showcase how the regularizer can help attain desirable properties, such as fairness and load balancing, and present a dual online subgradient descent algorithm for solving this problem with low regret. Future directions include extending the results in this work to more general input models (e.g., non-stationary stochastic inputs and adversarial inputs).

References

Shipra Agrawal and Nikhil R. Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, page 1405–1424, USA, 2015. Society for Industrial and Applied Mathematics.

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

Alessandro Arlotto and Itai Gurvich. Uniformly bounded regret in the multisecretary problem. *Stochastic Systems*, 9(3):231–260, 2019.

Yossi Azar, Niv Buchbinder, and Kamal Jain. How to allocate goods in an online market? In *European Symposium on Algorithms*, pages 51–62. Springer, 2010.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *J. ACM*, 65(3), March 2018.

Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. Dual mirror descent for online allocation problems. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 613–628. PMLR, 13–18 Jul 2020.

Santiago R Balseiro, Jon Feldman, Vahab Mirrokni, and Shan Muthukrishnan. Yield optimization of display advertising with ad exchange. *Management Science*, 60(12):2886–2907, 2014.

Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 31–40, 2006.

Mohammad Hossein Bateni, Yiwei Chen, Dragos Ciocan, and Vahab Mirrokni. Fair resource allocation in a volatile marketplace. *Available at SSRN 2789380*, 2018.

Dimitri P Bertsekas. *Convex optimization theory*. Athena Scientific Belmont, 2009.

Dimitris Bertsimas, Vivek F Farias, and Nikolaos Trichakis. The price of fairness. *Operations research*, 59(1):17–31, 2011.

Dimitris Bertsimas, Vivek F Farias, and Nikolaos Trichakis. On the efficiency-fairness trade-off. *Management Science*, 58(12): 2234–2250, 2012.

Gong Chen and Marc Teboulle. Convergence analysis of a proximal-like minimization algorithm using bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993.

Nikhil R. Devanur and Thomas P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, EC '09, pages 71–78. ACM, 2009.

Nikhil R. Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. *J. ACM*, 66(1), jan 2019.

Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

Reza Eghbali and Maryam Fazel. Designing smoothing functions for improved worst-case competitive ratio in online optimization. In *Advances in Neural Information Processing Systems*, pages 3287–3295, 2016.

Jon Feldman, Nitish Korula, Vahab Mirrokni, S. Muthukrishnan, and Martin Pál. Online ad assignment with free disposal. In *Proceedings of the 5th International Workshop on Internet and Network Economics*, WINE '09, pages 374–385. Springer-Verlag, 2009.

Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In *Proceedings of the 18th annual European conference on Algorithms: Part I*, ESA'10, pages 182–194. Springer-Verlag, 2010.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Stefanus Jasin. Performance of an lp-based control for revenue management with unknown demand parameters. *Operations Research*, 63(4):909–915, 2015.

Rodolphe Jenatton, Jim Huang, Dominik Csiba, and Cedric Archambeau. Online optimization and regret guarantees for non-additive long-term constraints. *arXiv preprint arXiv:1602.05394*, 2016.

Xiaocheng Li and Yinyu Ye. Online linear programming: Dual convergence, new algorithms, and regret bounds. 2019.

Xiaocheng Li, Chunlin Sun, and Yinyu Ye. Simple and fast algorithm for binary integer and online linear programming. *arXiv preprint arXiv:2003.02513*, 2020.

Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5):22–es, October 2007. ISSN 0004-5411.

Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on networking*, 8 (5):556–567, 2000.

John F Nash Jr. The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162, 1950.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Xiaoqi Tan, Bo Sun, Alberto Leon-Garcia, Yuan Wu, and Danny HK Tsang. Mechanism design for online resource allocation: A unified approach. In *Abstracts of the 2020 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*, pages 11–12, 2020.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 928–935. AAAI Press, 2003.