Regularized Online Allocation Problems: Fairness and Beyond

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

[1]Columbia University [2]University of Chicago [3]Google Research

Abstract

Problem setting: requests repeatedly arrive over time and, for each request, a decision maker needs to take an action that generates a reward and consumes resources.

Our contribution: we introduce the *regularized online allocation problem*, a variant of online allocation problem that includes a non-linear regularizer acting on the total resource consumption

Our Goal: The objective is to simultaneously maximize total rewards and the value of the regularizer subject to the resource constraints.

Problem Formulation

- Finite horizon with T time periods
- Resources $j = 1 \ldots m$ with capacities $\rho_j T$
- **Online Process**: At a specific time period t:
  - A request arrives with revenue function f_t, and consumption matrix b_t
  - Choose action x_t from action space $\mathcal{X}$
  - Generates a revenue $f_t(x_t)$
  - Consumes resources $b_t x_t$
- **Challenge**: Information about the future requests is not known in advance
- **Offline problem**:

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

  where r is a concave regularizer on the consumption.
- **Regularizer**: While maximizing the total revenue, we also would like to impose certain properties of the allocation, such as fairness among advertisers, load balancing for machine scheduling, etc.

Examples of the Regularizer r

- **Max-min fairness on the consumption**: $r(s) = \lambda \min_j (s_j/\rho_j)$
- **Max-min fairness on the reward (Santa Claus Fairness)**: we can introducing auxiliary budget constraint on the reward and use the above fairness on the consumption
- **Load balancing** on the consumption: $r(s) = -\lambda \min_j (s_j/\rho_j)$
- **Penalty** when under-delivering: $r(s) = -\lambda \sum_j c_j \max(s_j - t_j, 0)$
- **Penalty** when consumption is above a certain level: $r(s) = -\lambda \sum_j c_j \max(t_j - s_j, 0)$

Lagrangian Duality to the Rescue

Introducing Lagrange multipliers $\mu \ge 0$ for resource constraints $\frac{1}{T}\sum_{t=1}^{T} b_t x_t = a \le \rho$ where a is an auxiliary variable (target consumption), we obtain the dual problem:

$$D(\mu)$$
$$= \max_{x_t \in \mathcal{X}, a \le \rho} \sum_{t=1}^{T} f_t(x_t) + T \cdot r(a) + \mu^\top \left(Ta - \sum_{t=1}^{T} b_t x_t\right)$$
$$= \sum_{t=1}^{T} \left(\max_{x_t \in \mathcal{X}} \{f(x_t) - \mu^\top b_t x_t\}\right) + T \max_{a \le \rho} \{r(a) + \mu^\top a\}$$

Challenge # 1: How do we make decisions and update the target consumption?

- If "optimal" dual variables μ^* are known, we can take actions to maximize the dual-variable adjusted reward and update the target consumption:

$$x_t = \underset{x \in \mathcal{X}_t}{\operatorname{argmax}} \left\{f_t(x) - (\mu^*)^\top b_t x\right\}$$
$$a_t = \underset{a \le \rho}{\operatorname{argmax}} \{r(a) + (\mu^*)^\top a\}$$

- ...but we DO NOT know μ^* in advance

Challenge #2: How do we compute good dual variables?

- We can obtain an online subgradient of the dual function
- Compute dual variables by minimizing the dual function using subgradient descent

Our Algorithm: Dual Subgradient Descent Algorithm

Input: Initial dual solution μ_0, total number of time periods T, initial resources $B_0 = T\rho$, weight vector $w \in \mathbb{R}^m_{++}$, and step-size η.

for $t = 0, \ldots, T-1$ **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 dual problem:

$$g_t = -b_t \tilde{x}_t + a_t .$$

Update the dual variable by weighted, projected subgradient descent:

$$\mu_{t+1} = \arg\min_{\mu \in \mathbb{R}^m_+} \langle g_t, \mu \rangle + \frac{1}{2\eta} \|\mu - \mu_t\|_w^2 .$$

end

Theoretical Results (Informally)

- Let P be unknown i.i.d. distribution of requests
- Suppose that resources B are proportional to number of time periods T
- The regret of online dual gradient descent satisfies

$$\text{Regret(ALG)} = \sup_P \mathbb{E}_P[\text{OFFLINE} - \text{ALG}] \le O(T^{1/2}),$$

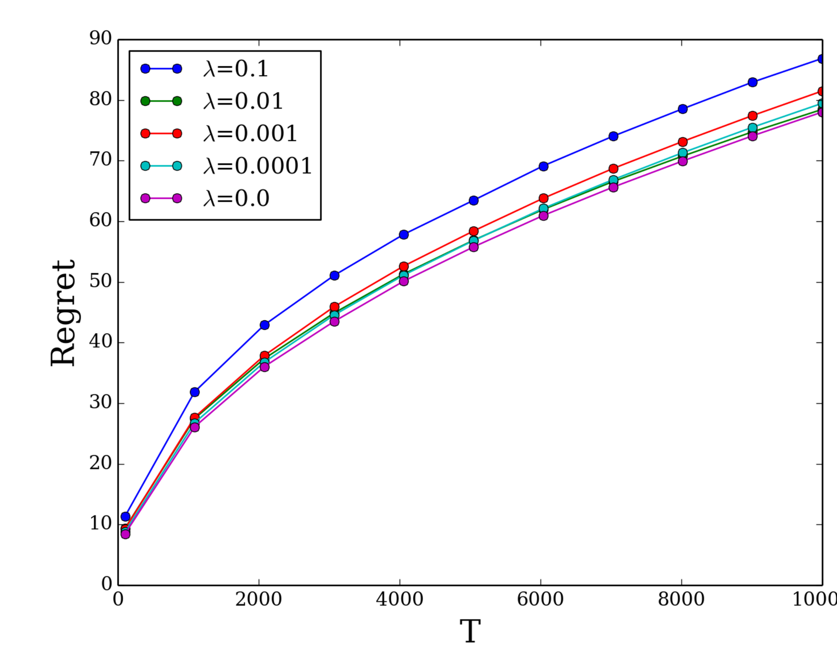
where
  - ALG = reward with regularizer collected by the proposed online algorithm
  - OFFLINE = maximal reward with regularizer from the offline problem

Numerical Experiments

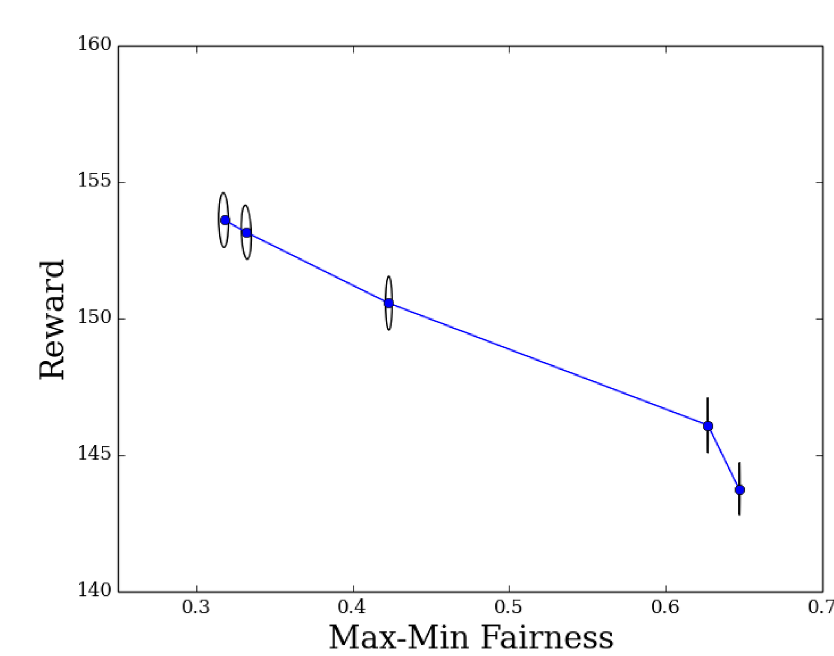
- **Problem setting:** A publisher has agreed to deliver ad slots (the requests) to different advertisers (the resources) to maximize the cumulative click-through rates (the reward) of the assignment.
- **Regularized Online Problem:** The goal is to design an online allocation algorithm that maximizes the total expected click-through rate with a max-min fairness regularizer on resource consumption:

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

where λ is the weight of the regularizer.



Regret versus horizon T with different regularization levels λ.



Reward $\sum_{t=1}^{T} q_t x_t$ versus max-min fairness $\min_j \left(\sum_{t=1}^{T} (x_t)_j / T\rho_j\right)$.

Fairness can be significantly improved with a small amount of click-through rate reduction!