Playing in the Dark: No-Regret Learning with Online Constraints

Abhishek Sinha

Joint work with Rahul Vaze

Tata Institute of Fundamental Research

Mumbai, India

December 17, 2023

The world around us is uncertain ...

... and full of surprising constraints

Charged with uncertainty - EV charging infra

Published 05 Feb 2019 **Last Updated** 05 Mar 2019 18:07

Arran Brown & Alexandra Dockreay Contact Author

Tags Transport Europe North America

Share: **Export:**

Skip to: Article Snapshot

The New York Times

Massive Power Failure Sweeps Across Italy

Share full article

By Elisabetta Povoledo, International Herald Tribune

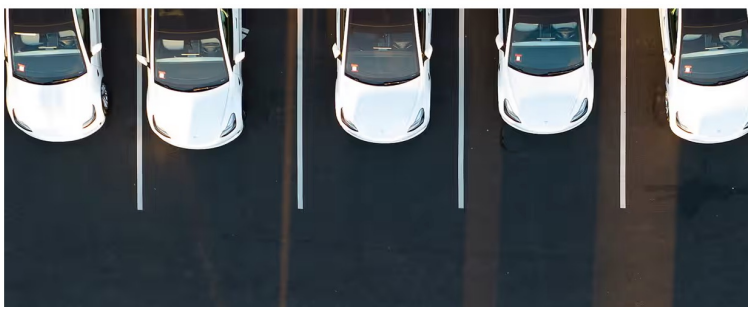
Sept. 28, 2003

International Herald Tribune

ROME, Sept. 28 — A power failure left most of Italy without

Tesla behind eight-vehicle crash was in 'full self-driving' mode, says driver

San Francisco crash is the latest in a series of accidents blamed on Tesla technology, which is facing regulatory scrutiny



Optimization in the dark

Problem: Solve a constrained convex optimization problem

$$\min_{x \in \Omega^*} f(x)$$

when we know

(1) neither the objective function f
(2) nor the constraint set Ω^*

Optimization in the dark

Problem: Solve a constrained convex optimization problem

$$\min_{x \in \Omega^*} f(x)$$

when we know

(1) neither the objective function f
(2) nor the constraint set Ω^*

- Clearly an impossible task in the one-shot setting when both f and Ω^* can be selected adversarially.

Optimization in the dark

Problem: Solve a constrained convex optimization problem

$$\min_{x \in \Omega^*} f(x)$$

when we know

(1) neither the objective function f
(2) nor the constraint set Ω^*

- Clearly an impossible task in the one-shot setting when both f and Ω^* can be selected adversarially.

Question: What can we say about the online setting?

Online Optimization in the dark

- Let $\{f_t : \Omega \to \mathbb{R}\}_{t=1}^{T}$ be a sequence of convex functions and Ω be a convex action set
- Consider the following protocol:

Online Optimization in the dark

- Let $\{f_t : \Omega \to \mathbb{R}\}_{t=1}^{T}$ be a sequence of convex functions and Ω be a convex action set
- Consider the following protocol:
  1. On every round $t \in [T]$, a policy selects a feasible action $x_t \in \Omega$

Online Optimization in the dark

- Let $\{f_t : \Omega \to \mathbb{R}\}_{t=1}^{T}$ be a sequence of convex functions and Ω be a convex action set
- Consider the following protocol:
  1. On every round $t \in [T]$, a policy selects a feasible action $x_t \in \Omega$
  2. The adversary reveals the cost function $f_t : \Omega \to \mathbb{R}$ and a *constraint* function $g_t : \Omega \to \mathbb{R}$

Online Optimization in the dark

- Let $\{f_t : \Omega \to \mathbb{R}\}_{t=1}^{T}$ be a sequence of convex functions and Ω be a convex action set
- Consider the following protocol:
  1. On every round $t \in [T]$, a policy selects a feasible action $x_t \in \Omega$
  2. The adversary reveals the cost function $f_t : \Omega \to \mathbb{R}$ and a *constraint* function $g_t : \Omega \to \mathbb{R}$
  3. $\Omega^* \subseteq \cap_{t \geq 1}\{x : g_t(x) \leq 0\}, \quad \Omega^* \neq \emptyset$

Informal Goal: Choose action sequence $\{x_t\}_{t \geq 1}$ which (1) minimizes the cumulative cost $\sum_{t=1}^{T} f_t(x_t)$ and (2) satisfies the constraints as tightly as possible, i.e., $g_t(x_t) \precsim 0, \forall t$.

Example: The Hidden Set Problem

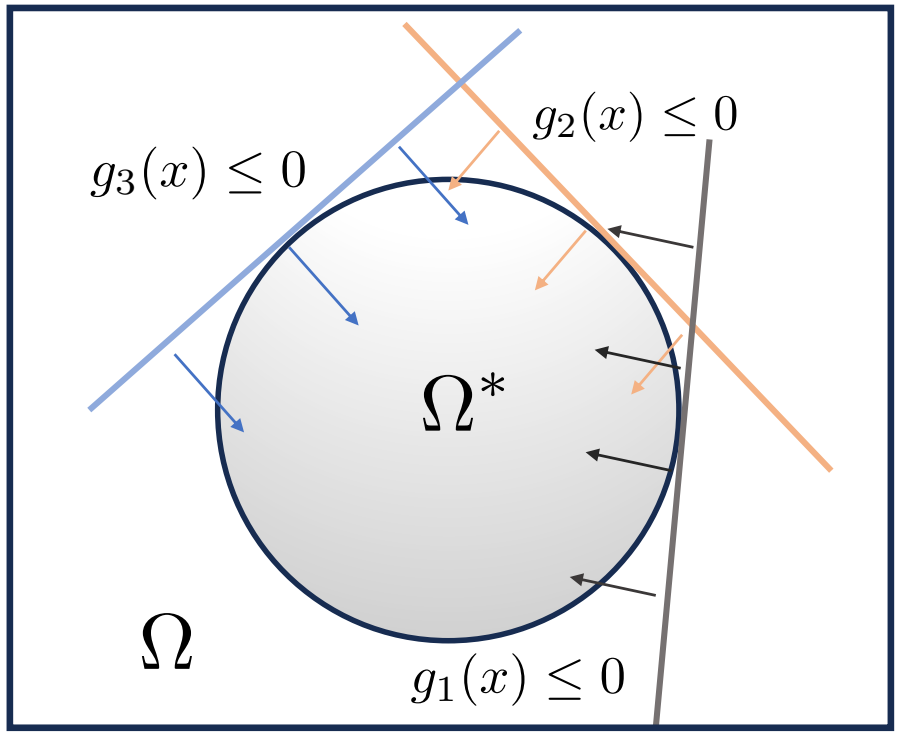


Figure: Optimization over a hidden constraint set Ω^*. On every round t, the adversary reveals a convex cost function f_t (not shown) and a supporting hyperplane to Ω^*. The objective is to do as well as the optimal point in Ω^* in terms of the cumulative cost.

Content

- Online Constraint Satisfaction
- Generalized OCO
- Bandits with Constraints

Recap: Online Convex Optimization (OCO)

Consider a repeated game. On round t:

1. An online policy chooses a feasible action $x_t \in \Omega$
2. The adversary chooses a convex cost function $f_t : \Omega \to \mathbb{R}$
3. The policy incurs the cost $f_t(x_t)$ and the function f_t (or, just the gradient $\nabla f_t(x_t)$) is revealed to the policy

The objective of the policy is to minimize the *Regret*

$$\text{Regret}_T = \sup_{x^* \in \Omega} \Big(\underbrace{\sum_{t=1}^{T} f_t(x_t)}_{\text{cost of the policy}} - \underbrace{\sum_{t=1}^{T} f_t(x^*)}_{\text{cost of the fixed benchmark}} \Big).$$

- There is no constraint functions

Algorithms for OCO

- Many online algorithms guarantee sublinear regret in various settings.
- Examples include Follow-the-regularized-leader (FTRL) and Online Mirror descent (OMD)
- In this talk we will particularly focus on simple Online Gradient Descent (OGD) class of policies with adaptive step sizes

Algorithm 1 Online Gradient Descent (OGD)

Input: Non-empty closed convex set $\Omega \subseteq \mathbb{R}^d$, Sequence of convex cost functions $\{f_t\}_{t\geq 1}$, step sizes $\eta_1, \eta_2, \ldots, \eta_T > 0$, Projection oracle $\mathcal{P}_\Omega(\cdot)$ onto the set Ω

1: Initialize x_1 arbitrarily
2: **for** each round $t \geq 1$ **do**
3: Predict x_t, observe f_t, incur a cost of $f_t(x_t)$.
4: Compute a (sub)-gradient ∇_t of f_t at x_t. Let $G_t = ||\nabla_t||_2$.
5: Update $x_{t+1} = \mathcal{P}_\Omega(x_t - \eta_t \nabla_t)$.
6: **end for**

Adaptive Regret Bounds for OGD

1. (**Convex Costs** [Duchi et al., 2011]) Setting $\eta_t \leftarrow \frac{D}{\sqrt{2\sum_{\tau=1}^{t-1} G_\tau^2}}$, $t \geq 1$, OGD (a.k.a. AdaGrad) achieves

$$\text{Regret}_T \leq D\sqrt{2\sum_{t=1}^{T} G_t^2}, \tag{1}$$

   where $\text{Diam}(\Omega) = D$.

2. (**Strongly-convex Costs** [Hazan et al. 2007]) Setting $\eta_t \leftarrow \frac{1}{\sum_{s=1}^{t-1} H_s}$, $t \geq 1$, OGD achieves

$$\text{Regret}_T \leq \frac{1}{2}\sum_{t=1}^{T} \frac{G_t^2}{\sum_{s=1}^{t} H_s}, \tag{2}$$

   where H_t is the strong convexity parameter of the function f_t.

Our Result

A universal blackbox reduction

$$\text{Constrained online problem} \implies \text{Standard OCO problem.}$$

Our Result

A universal blackbox reduction

$$\text{Constrained online problem} \implies \text{Standard OCO problem.}$$

1. Our reduction is policy agnostic - works with any *off-the-shelf* adaptive OCO policy
   - The details of the algorithm is irrelevant - it could be taken to be FTRL, FTPL, OMD as convenient
2. Yields the optimal regret and cumulative violation bounds
3. Connection with Queueing theory - network control policy with adversarial arrival and departure process

Part I: Online Constraint Satisfaction (OCS)

- We begin with the simpler constraints-only case with no cost function (*i.e.*, $f_t = 0, \forall t.$)
- On round t, after the policy selects its action $x_t \in \Omega$, the adversary reveals a vector of k convex and Lipschitz functions

$$(g_{t,1}(x), g_{t,2}(x), \ldots, g_{t,k}(x)).$$

- The goal is to keep the cumulative constraint violation over any interval for each component small, *i.e.*,

$$\mathbb{V}_T \equiv \max_{\mathcal{I} \subseteq [T]} \max_{j=1}^{k} \sum_{t \in \mathcal{I}} g_{t,j}(x_t) = o(T).$$

  - An online version of the multi-objective control problem

Application: Online Multi-task Learning

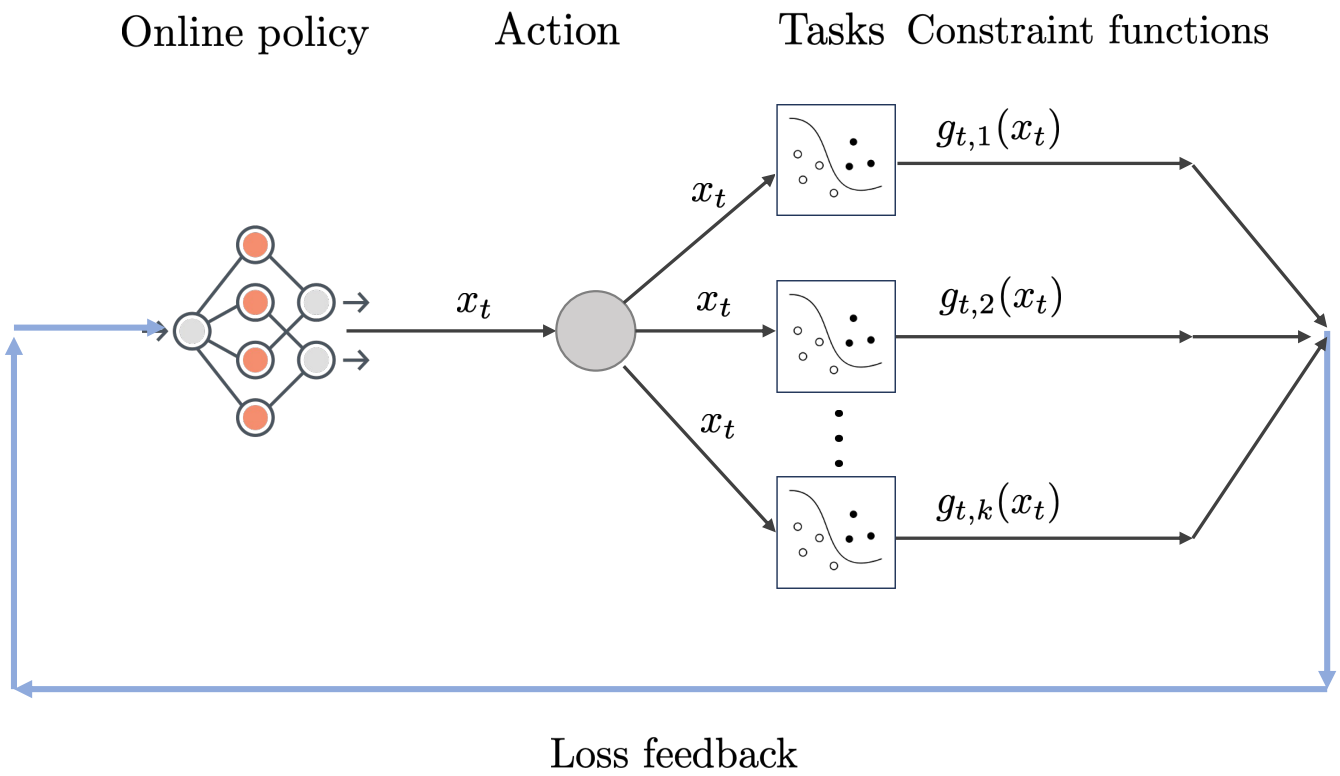


Figure: A schematic of online multi-task learning problem.

- The j^{th} task on round t is completed if $g_{t,j}(x_t) \leq 0,\ j \in [k]$.

Building intuition - Take 1, Duality

- To build up intuition, consider the following time-invariant case where $f_t = f$, and $g_t = g, \forall t \geq 1$. It is equivalent to the following offline problem:

$$\min_{x \in \Omega} f(x), \text{ s.t. } g(x) \leq 0.$$

- Assuming strong duality, the above problem is equivalent to solving the following saddle point problem

$$\min_{x \in \Omega} \max_{\lambda \geq 0} L(x, \lambda),$$

where $L(x, \lambda) = f(x) + \lambda g(x)$ is the associated Lagrangian.

- The saddle point problem is equivalent to computing a Nash Equilibrium of a two player zero-sum game with payoff function $L(x, \lambda)$.

Building intuition - Take 1, Duality

- Now, a Nash Equilibrium for a two-player zero-sum game can be computed iteratively by running a regret minimizer for each player.
- [Daskalakis et al. 2021] showed that adaptivity helps - yields a faster $\mathrm{poly}(\log T)$ convergence in multi-player zero sum games!

Building intuition - Take 1, Duality

- Now, a Nash Equilibrium for a two-player zero-sum game can be computed iteratively by running a regret minimizer for each player.
- [Daskalakis et al. 2021] showed that adaptivity helps - yields a faster $\mathrm{poly}(\log T)$ convergence in multi-player zero sum games!

Consider the following algorithm:

(1) The x-player runs an adaptive OGD with the cost function

$$L(x, \lambda_t) = f(x) + \lambda_t g(x).$$

(2) The λ-player runs an OGD with a constant (unit) step-size

$$\lambda_t = \left(\lambda_{t-1} + g(x_{t-1})\right)^+.$$

Building intuition - Take 1, Duality

Next, consider a natural extension of the above algorithm for time-varying cost and constraint functions:

(1) The x-player runs an adaptive OGD with the cost function

$$L_t(x, \lambda_t) = f_t(x) + \lambda_t g_t(x).$$

(2) The λ-player runs an OGD with a constant (unit) step-size

$$\lambda_t = \left(\lambda_{t-1} + g_t(x_t)\right)^+.$$

- λ_t depends on x_t.

We will show that the above algorithm indeed achieves optimal regret and violation bounds.

Building intuition, Take 2 - Stability

The following analogy would be helpful to visualize the algorithm

- Keep track of the violations through queueing recursions

$$Q_i(t) = \big(Q_i(t-1) + \underbrace{g_{t,i}(x_t)}_{\text{violation for round } t}\big)^+, \;\; Q_i(0) = 0, \;\; \forall i$$

$$\lambda_i(t) - \mu_i(t) = g_{t,i}(x_t)$$

- Intuitively, stabilizing the queues implies that the long-term target rates are satisfied

Stabilization via Drift Minimization

Define the quadratic Lyapunov function $\Phi(t) = \sum_i Q_i^2(t)$. Its drift is upper bounded as

$$\Phi(t) - \Phi(t-1) \leq 2 \sum_{i=1}^{k} Q_i(t) g_{t,i}(x_t).$$

Stabilization via Drift Minimization

Define the quadratic Lyapunov function $\Phi(t) = \sum_i Q_i^2(t)$. Its drift is upper bounded as

$$\Phi(t) - \Phi(t-1) \leq 2 \sum_{i=1}^{k} Q_i(t) g_{t,i}(x_t).$$

Define the surrogate cost function as

$$\hat{f}_t(x) \equiv 2 \sum_{i=1}^{k} Q_i(t) g_{t,i}(x).$$

Note that the coefficients $\vec{\boldsymbol{Q}}(t)$ depend on the past and current actions. We now propose

> OCS Policy: Run any adaptive OCO policy Π on the surrogate cost functions $\{\hat{f}_t\}_{t \geq 1}$.

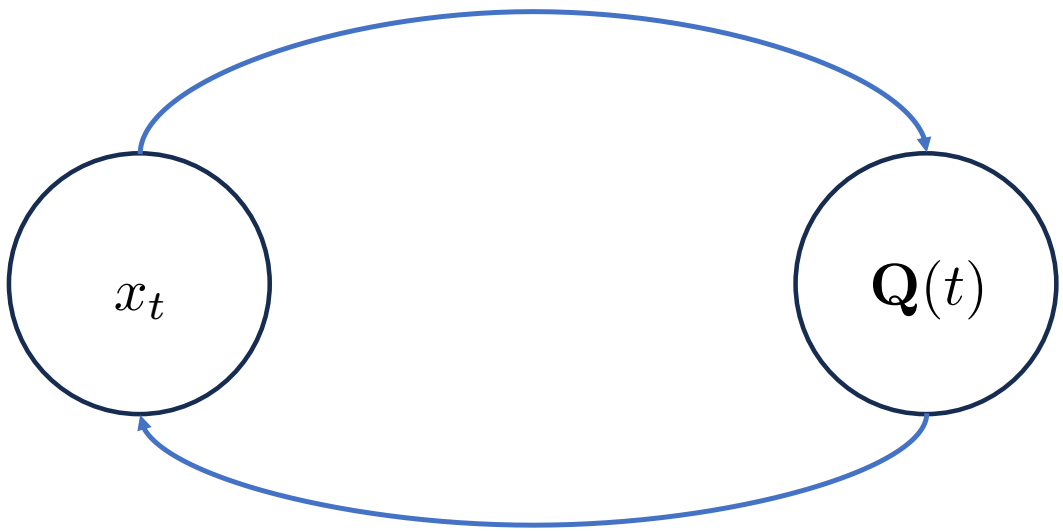
Analysis

The challenge is to account for the closed-loop interaction between

- The action $\boldsymbol{x}_t$ (which depends on the queue lengths via $\hat{f}_t$) and
- The queue-lengths $\boldsymbol{Q}(t)$ (which depends on the action)

through the policy Π

$$Q_i(t) = (Q_i(t-1) + g_{t,i}(x_t))^+$$



$$\Pi \leftarrow \hat{f}_t(x) \equiv \langle \mathbf{Q}(t), \mathbf{g}_t(x) \rangle$$

Interaction between action and queue lengths

Regret decomposition

Let $\boldsymbol{x}^* \in \Omega^*$ be any fixed action that satisfies all constraints. Then

$$\begin{aligned}
\Phi(\tau) - \Phi(\tau-1) &\leq 2\sum_i Q_i(\tau) g_{\tau,i}(x_\tau) \\
&\overset{g_{\tau,j}(x^*)\leq 0}{\leq} 2\sum_i Q_i(\tau) g_{\tau,i}(x_\tau) - 2\sum_i Q_i(\tau) g_{\tau,i}(x^*) \\
&= \underbrace{\hat{f}_\tau(x_\tau)}_{\text{surrogate cost}} - \hat{f}_\tau(x^*)
\end{aligned}$$

Summing up, for any $t \geq 1$, we have

$$\sum_i Q_i^2(t) \equiv \Phi^2(t) - \Phi^2(0) \leq \underbrace{\text{Regret}'_t}_{\text{surrogate regret - depends on } \{\boldsymbol{Q}(\tau)\}_{1\leq\tau\leq t}} \tag{3}$$

- a non-linear recurrence that we need to solve

Violation bounds I : Convex functions

Direct calculations yield

$$G_t^2 \equiv ||\nabla \hat{f}_t(x_t)||^2 \leq kG^2 \sum_i Q_i^2(t).$$

Substituting the above in the adaptive regret bound Eq. (1) for OGD yields

$$\sum_i Q_i^2(t) \leq GD\sqrt{2k}\sqrt{\sum_{\tau=1}^{t}\sum_i Q_i^2(\tau)}, \quad \forall t \geq 1.$$

Solving the above non-linear recurrence (a Math puzzle!), we have

$$\mathbb{V}_T \leq \max_i Q_i(T) = O(\sqrt{T}).$$

■

Violation bounds II : Strongly Convex functions

In this case, the strong-convexity parameter for the t^{th} surrogate function is $H_t = \alpha Q(t)$. Hence, the adaptive regret bound (2) for OGD yields

$$\sum_i Q_i^2(t) \leq \frac{kG^2}{4\alpha} \sum_{\tau=1}^{t} \frac{\sum_{i=1}^{k} Q_i^2(\tau)}{\sum_{s=1}^{\tau} \sum_{i=1}^{k} Q_i(s)}, \quad t \geq 1.$$

Solving this non-linear recurrence inequality (a bit harder puzzle!) yields the following violation bound

$$\mathbb{V}_T \leq \max_i Q_i(T) = O(\log T).$$

■

Generalized benchmark: S-feasibility

- Requiring the existence of a feasible action that satisfies all constraints on all rounds could be restrictive
- Can relax this assumption by requiring that sum of the constraint functions over any interval of length S is non-positive

Theorem: For convex constraints, the same adaptive OGD policy achieves the following violation bound

$$\mathbb{V}_i(T) = O(\max(\sqrt{ST}, S)), \ \forall i.$$

- The policy is oblivious to the value of S.

Part II: The Generalized OCO Problem

- On every round t, a convex cost function $f_t : \Omega \to \mathbb{R}$ and a convex constraint function $g_t : \Omega \to \mathbb{R}$ is revealed.
- Recall that Ω^* is the set of feasible actions that satisfies all constraints. Our objective is to design an online policy that yields

$$\begin{aligned} \text{Regret}_T &\equiv \sup_{x^* \in \Omega^*} \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) = o(T), \\ \textit{and} \quad \mathbb{V}_T &\equiv \sum_{t=1}^{T} \max(0, g_t(x_t)) = o(T). \end{aligned}$$

- Preprocessing: $g_t \leftarrow \max(0, g_t), \quad \forall t \geq 1.$

  - $Q(t)$ simply becomes the cumulative sum of the constraint violations

The Drift-Plus-Penalty Framework

Originally proposed by [Neely 2010] for stochastic network optimization and generalizes the MAX-WEIGHT policy.

The Drift-Plus-Penalty Framework

Originally proposed by [Neely 2010] for stochastic network optimization and generalizes the MAX-WEIGHT policy. Strategy:

Minimize the drift-plus-penalty, i.e., for some fixed $V > 0$, we define the surrogate cost function on round t as:

$$\hat{f}_t(x) \equiv \underbrace{2Q(t)g_t(x)}_{\text{drift upper bound}} + \underbrace{Vf_t(x)}_{\text{penalty}}.$$

Generalized OCO Policy: Run any adaptive OCO policy Π on the surrogate cost functions $\{\hat{f}_t\}_{t \geq 1}$.

Generalized Regret Decomposition

Let $\boldsymbol{x}^* \in \Omega^*$ be any fixed action that satisfies all constraints. We have

$$\begin{aligned} & \Phi(\tau) - \Phi(\tau-1) + V(f_\tau(x_\tau) - f_\tau(x^*)) \\ \overset{g_\tau(x^*)\le 0}{\le} \quad & (Vf_\tau(x_\tau) + 2Q(\tau)g_\tau(x_\tau)) - (Vf_\tau(x^*) + 2Q(\tau)g_\tau(x^*)) \\ = \quad & \hat{f}_\tau(x_\tau) - \hat{f}_\tau(x^*). \end{aligned}$$

Summing up, we have

$$Q^2(t) + V \underbrace{\mathrm{Regret}_t(x^*)}_{\text{regret for costs}} \le \underbrace{\mathrm{Regret}'_t}_{\text{surrogate - depends on } \{\boldsymbol{Q}(\tau)\}_{1\le\tau\le t}}, \qquad \forall t \ge 1 \quad (4)$$

Regret and violation bounds - Convex costs

The adaptive regret bound for the OGD applied to the surrogate costs leads to the following system of recursive inequalities:

$$Q^2(t) + V\mathrm{Regret}_t(x^*) \leq 2GD\sqrt{\sum_{\tau=1}^{t} Q^2(\tau)} + 2GDV\sqrt{t}.$$

Solving this recursion, we have:

Theorem 1 (SV'23)

Setting $V = \sqrt{T}$, for convex cost functions, we have

$$\mathrm{Regret}_t = O(\sqrt{t}), \text{ and } \mathbb{V}(t) = O(T^{3/4}).$$

The latter can be improved to $\mathbb{V}_t = O(\sqrt{T})$ when $\mathrm{Regret}_t \geq 0$.

Regret and violation bounds - Strongly Convex costs

Similarly, the adaptive regret bound for strongly convex costs yields

$$Q^2(t) + V\mathrm{Regret}_t(x^*) \leq \frac{VG^2}{\alpha}\ln(t) + \frac{G^2}{\alpha V}\sum_{\tau=1}^{t}\frac{Q^2(\tau)}{\tau}.$$

Solving this recursion yields the following:

Theorem 2 (SV'23)

Setting $V = \frac{2G^2\ln(T)}{\alpha}$, for strongly convex costs, we have

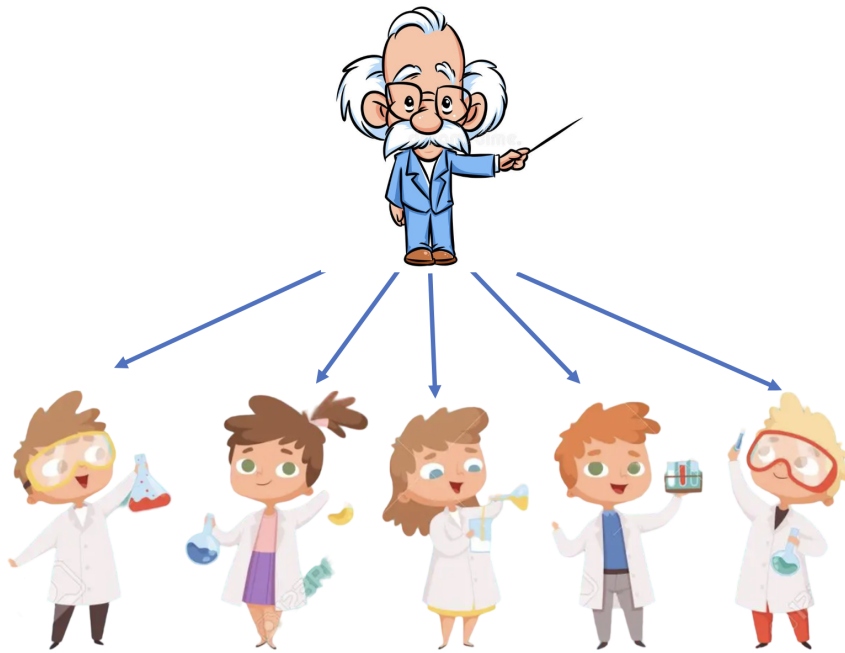
$$\mathrm{Regret}_t = O(\ln t), \text{ and } \mathbb{V}(t) = O(\sqrt{t\log T/\alpha}).$$

The latter can be improved to $\mathbb{V}_t = O(\log(T)/\alpha)$ when $\mathrm{Regret}_t \geq 0$.

The second part is surprising - logarithmic violation bound even for non strongly convex constraint functions!

Part III: Fair Allocation

- A Professor (or a grant agency) receives grants for research projects sequentially over time and wants to assign them *fairly* to his N PhD students (or PIs)



p_1 p_2 p_3 p_4 p_5

Probabilities for successfully completing the projects

- The skill levels of the students are initially unknown to the Professor. It can only be learned from their past performances.

Setup

- The students remain in the school for T years
- To meet the degree requirements, student i must solve on the average at least λ_i problems per year
  - The minimum requirements are known to be feasible in expectation
- The advisor wants to solve as many problems as possible subject to each student meeting the minimum requirements

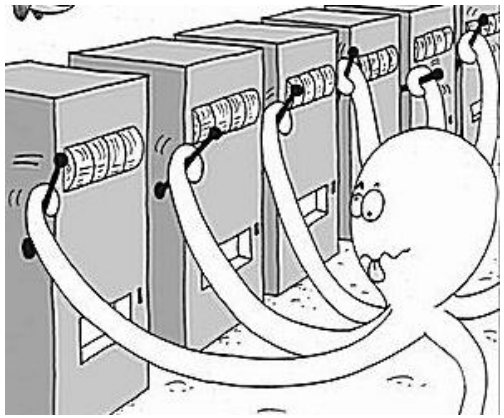
> Question: What is an optimal problem assignment strategy for the Professor?

Applications: Recommendation systems, Online ad allocation, Crowdsourcing, Wireless scheduling, ...

The Fair Bandit Problem

- We are given a set of N arms and a target reward rate vector $\boldsymbol{\lambda} \in \mathbb{R}_N^+$
- On round t, an online policy π selects an arm $I_t \in [N]$ to play
- Both the policy and the selected arm I_t receive an i.i.d. random reward of value $r_{I_t}(t) \in [0, 1]$
- The goal of the policy π is to maximize the rewards while ensuring that the long-term reward accrual rate of arm i is at least $\lambda_i, \forall i \in [N]$.
- The problem is interesting in both Full-information and Bandit feedback settings

Recap: Stochastic Multi-Armed Bandits



- There is a set of N arms. A policy pulls one arm at a time.
- The reward $r_i(t)$ of the i^{th} arm is sampled i.i.d. from a distribution in $[0, 1]$ with an unknown mean $\mu_i, i \in [N]$.
- The objective is to sequentially pull the arms $\{I_t\}_{t=1}^{T}$ so as to obtain a sublinear pseudo-regret:

$$\text{Regret}_T \equiv T \max_i \mu_i - \mathbb{E} \sum_{t=1}^{T} \sum_i r_i(t) \mathbb{1}(I_t = i)$$

Difference with Stochastic MAB Problem

- The standard MAB problem aims to pull the most rewarding arms the maximum number of times while ignoring the rest.
- This strategy does not work in the `Fair Bandit` problem as the suboptimal arms receive negligible rewards
  - Instead of the best arm, the policy needs to learn the the best distribution of arms
- Note that, when $\vec{\lambda} = \mathbf{0}$, the `Fair Bandit` problem reduces to the usual stochastic MAB problem.

Difference with Stochastic MAB Problem

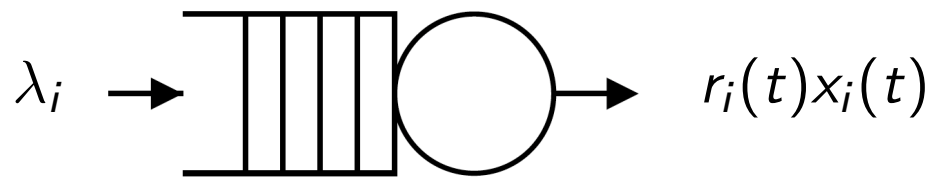
- The standard MAB problem aims to pull the most rewarding arms the maximum number of times while ignoring the rest.
- This strategy does not work in the `Fair Bandit` problem as the suboptimal arms receive negligible rewards
  - Instead of the best arm, the policy needs to learn the the best distribution of arms
- Note that, when $\vec{\lambda} = \mathbf{0}$, the `Fair Bandit` problem reduces to the usual stochastic MAB problem.

Main Result: Blackbox reduction to the adversarial MAB problem with simultaneously $O(T^{3/4})$ regret and cumulative violations.

Algorithmic ideas for BanditQ (Full-information setup)

- We keep track of the current rate of reward accruals through queueing recursions

$$Q_i(t) = \big(Q_i(t-1) + \underbrace{\lambda_i}_{\text{target rate}} - \underbrace{r_i(t)x_i(t)}_{\text{expected rate for round } t}\big)^+, Q_i(0) = 0, \forall i$$



- Intuitively, stabilizing the queues implies that the long-term target rates are satisfied
  - For us, only the rate stability of the queues will suffice, *i.e.*, $T^{-1}\mathbb{E}Q_i(T) \to 0, \forall i$.

Incorporating the Backlogs into Rewards

- To minimize the regret, we add a (scaled) reward of each arm to the queues and define a overall reward vector

$$r_i'(t) = (Q_i(t-1) + V) r_i(t), \ \forall i \in [N].$$

$V = \Theta(\sqrt{T})$ depends only on the horizon-length.

- This defines an instance of a MAB problem which we solve using

(Online Gradient Ascent Policy) The `BanditQ` policy selects the arms by running the OGA policy with adaptive step sizes:

$$\boldsymbol{x}(t) \leftarrow \mathrm{Proj}_{\Delta^N} \left[\boldsymbol{x}(t-1) + \frac{\boldsymbol{r}'(t-1)}{\sqrt{2 \sum_{\tau=1}^{t-1} ||\boldsymbol{r}'(\tau)||_2^2}} \right].$$

Bandit feedback

- With the bandit feedback, only the reward of the selected arm (*i.e.*, $r_{I_t}(t)$) is revealed.
- This leads to a slightly modified queueing recursion:

$$Q_i(t) = \left(Q_i(t-1) + \lambda_i - r_i(t)X_i(t)\right)^+,$$

  where $X_i(t) = \mathbb{1}(I_t = i)$.
- The OGA policy is replaced with an adversarial MAB policy [Putta and Agarwal 2022], which enjoys a second-order regret bound

$$\text{Regret} = \tilde{O}\left(\sqrt{N\sum_{t=1}^{T}||\boldsymbol{r}_t'||^2 + \max_{t\in[T]}||r_t'||_\infty\sqrt{NT}}.\right).$$

- The proof of the regret bound involves a similar self-bounding inequality as in the full-information case.

Results

- The second term of the bound, involving a max norm over the entire time horizon, requires a careful Martingale-based analysis

The following is our main result:

Theorem 3 (BanditQ Regret Bounds)
With $V = \sqrt{T}$, the BanditQ policy with bandit feedback achieves

$$\mathrm{Regret}(\boldsymbol{x}^*) = \tilde{O}(N^{5/4}T^{3/4}),\ \mathbb{V}(T) = O(N^{1/4}T^{3/4}).$$

Preliminary Experiments: Full-Information Setup

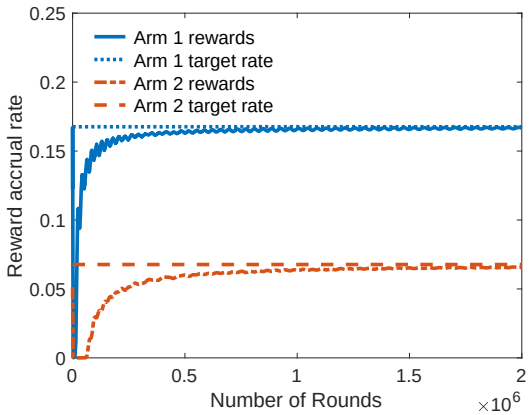


Figure: Reward accrual rates

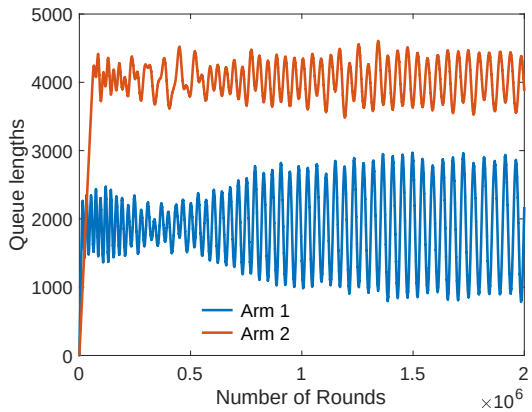


Figure: Queue lengths under `BanditQ`

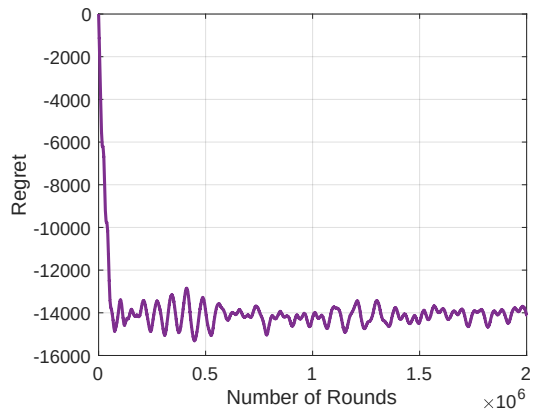


Figure: Regret of `BanditQ`

Preliminary Experiments: Bandit Information Setup

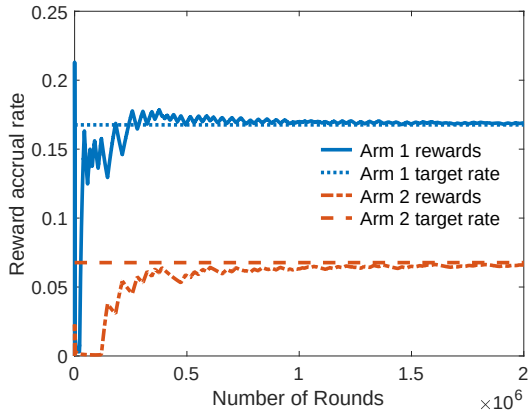


Figure: Reward accrual rates

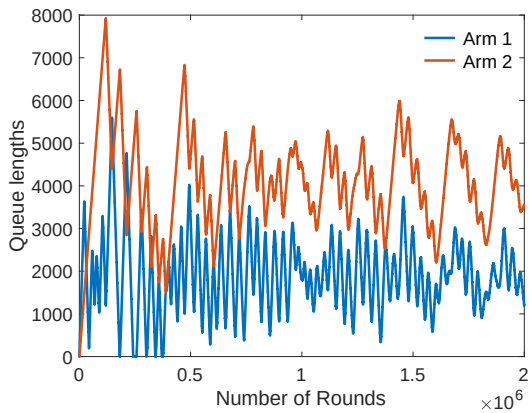


Figure: Queue lengths under `BanditQ`

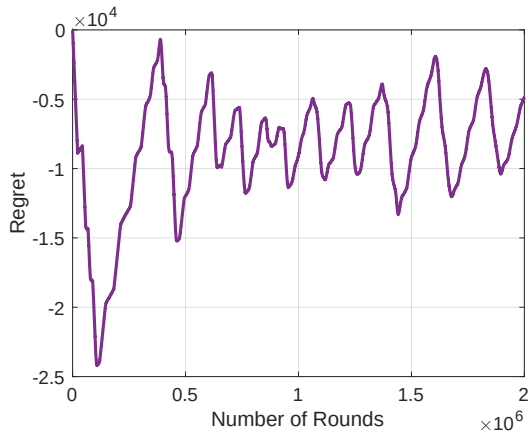


Figure: Regret of `BanditQ`

Open Problems

- Is it possible to derive instance-dependent logarithmic regret for `BanditQ`?
- How to extend the algorithm to the much more challenging `Bandit Convex Optimization` (BCO) Setup?
- Can we learn non-separable utility functions while simultaneously satisfying the long-term constraints?

References

1. Sinha, Abhishek, and Rahul Vaze. "Playing in the Dark: No-regret Learning with Adversarial Constraints." arXiv preprint arXiv:2310.18955 (2023).

2. Sinha, Abhishek. "`BanditQ`: Fair Multi-Armed Bandits with Guaranteed Rewards per Arm." arXiv preprint arXiv:2304.05219 (2023).

3. Sinha, Abhishek, Joshi, A., Bhattacharjee, R., Musco, C., Hajiesmaili, M. (2023). "No-regret Algorithms for Fair Resource Allocation." NeurIPS 2023.