No-regret Algorithms for Fair Resource Allocation

Abhishek Sinha [1]

joint work with

A. Joshi [1] R. Bhattacharjee [2] C. Musco [2] M. Hajiesmaili [2]

[1] **Tata Institute of Fundamental Research, Mumbai**

[2] UMass Amherst

Multi-user fair allocation

Problem: Four friends meet every year for their birthdays and share the cake by dividing it into four pieces. However, their current preferences for different pieces change with time, which they reveal after the cake is distributed. Devise a sequential strategy to divide the cakes **fairly** in the long run.

No-Regret Resource allocation

- N distinct resources to be shared by m users for multiple rounds.

No-Regret Resource allocation

- N distinct resources to be shared by m users for multiple rounds.
- On round t, the demand vector of the i^{th} agent is given by $\boldsymbol{x}_i(t)$ (which could be adversarially chosen) and its allocation by an online policy be denoted by $\boldsymbol{y}_i(t)$, resulting in a reward of

$$R_i(T) = \sum_{t=1}^{T} \langle \boldsymbol{x}_i(t), \boldsymbol{y}_i(t) \rangle, i \in [m].$$

No-Regret Resource allocation

- N distinct resources to be shared by m users for multiple rounds.
- On round t, the demand vector of the i^{th} agent is given by $\boldsymbol{x}_i(t)$ (which could be adversarially chosen) and its allocation by an online policy be denoted by $\boldsymbol{y}_i(t)$, resulting in a reward of

$$R_i(T) = \sum_{t=1}^{T} \langle \boldsymbol{x}_i(t), \boldsymbol{y}_i(t) \rangle, i \in [m].$$

- The standard learning theory prescribes algorithms (*e.g.*, Online gradient descent) for maximizing the sum of the rewards (*i.e.*, minimizing the regret), which is separable across time

$$\text{Regret}_T \equiv \underbrace{\sum_{i=1}^{m} R_i^*(T)}_{\text{Reward by a static offline policy}} - \underbrace{\sum_{i=1}^{m} R_i(T)}_{\text{Reward by an online policy}}.$$

No-Regret Resource allocation

- N distinct resources to be shared by m users for multiple rounds.
- On round t, the demand vector of the i^{th} agent is given by $\boldsymbol{x}_i(t)$ (which could be adversarially chosen) and its allocation by an online policy be denoted by $\boldsymbol{y}_i(t)$, resulting in a reward of

$$R_i(T) = \sum_{t=1}^{T} \langle \boldsymbol{x}_i(t), \boldsymbol{y}_i(t) \rangle, i \in [m].$$

- The standard learning theory prescribes algorithms (*e.g.*, Online gradient descent) for maximizing the sum of the rewards (*i.e.*, minimizing the regret), which is separable across time

$$\text{Regret}_T \equiv \underbrace{\sum_{i=1}^{m} R_i^*(T)}_{\text{Reward by a static offline policy}} - \underbrace{\sum_{i=1}^{m} R_i(T)}_{\text{Reward by an online policy}}.$$

Question: Is this a **fair** metric to optimize?

Example: Multi-user Caching

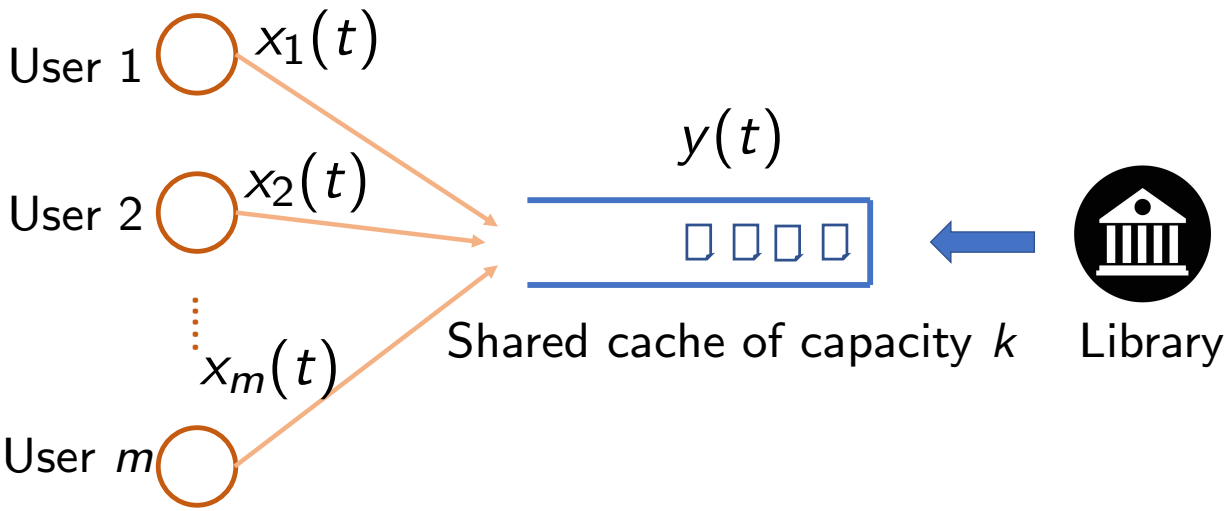


Figure: The online shared caching problem

Example: Multi-user Caching

- m users are connected to a single cache of capacity k.
- At the beginning of each round t, an online caching policy prefetches a set of k files $\boldsymbol{y}(t)$ such that

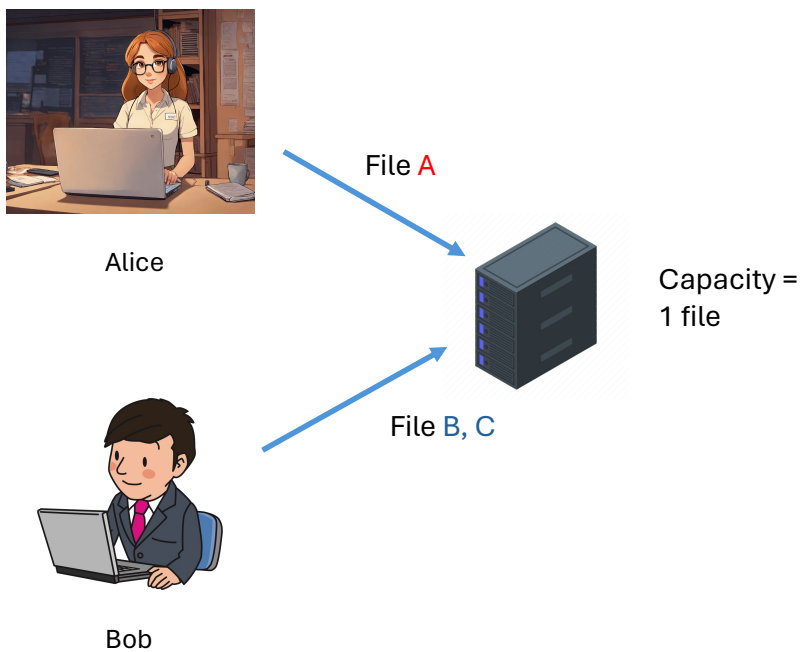
$$\sum_{j=1}^{N} y_j(t) = k, \quad y_j(t) \in \{0, 1\} \ , \forall j. \tag{1}$$

- File request of user i can be represented by a one-hot vector $\boldsymbol{x}_i(t)$ and hence, user i receives a hit of $\langle \boldsymbol{x}_i(t), \boldsymbol{y}(t) \rangle$ on round t.

> Question: What is a reasonable objective? Should we just maximize the cumulative hits across all users?

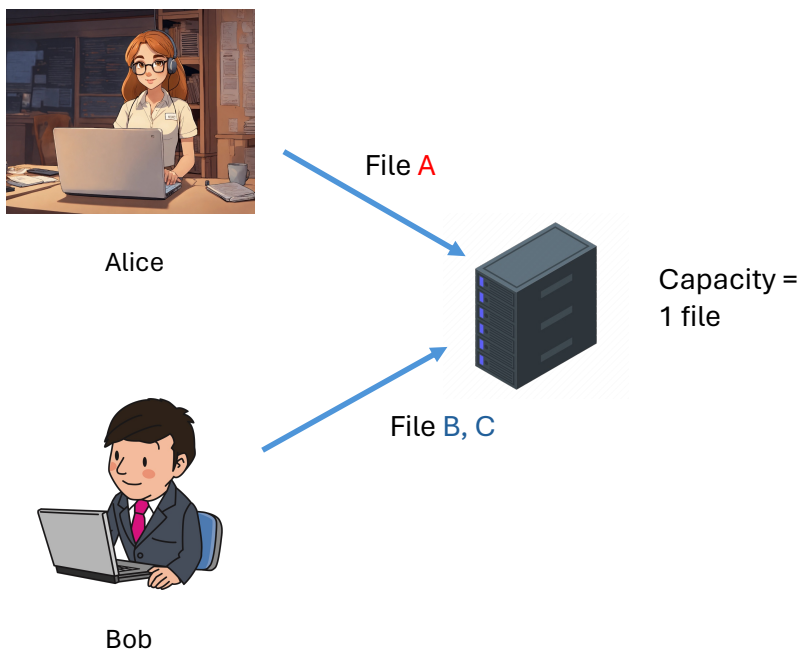
A case study

- Two users, Alice and Bob, are connected to a single cache of unit capacity ($k = 1$)
- On every round, Bob alternatively requests either file B or file C
- On the other hand, Alice always requests file A



A case study

- Two users, Alice and Bob, are connected to a single cache of unit capacity ($k = 1$)
- On every round, Bob alternatively requests either file B or file C
- On the other hand, Alice always requests file A



A static optimal policy maximizing the cumulative hit will always cache file A. This would yield 100% hit rate to Alice and 0% hit rate to Bob!

Unfairness with maximizing cumulative rewards

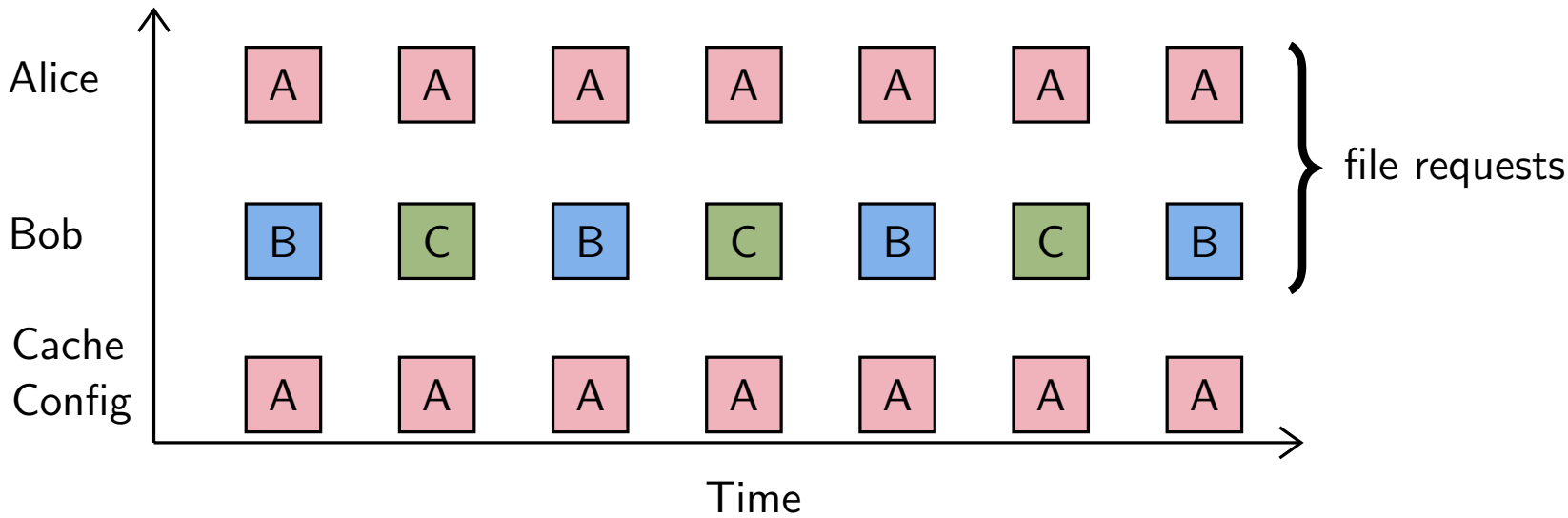


Figure: File requests from Alice and Bob

- Alice always requests file A. Bob requests files B or C alternatively. Only one file can be cached.
- Always caching file A will give us the highest possible cumulative hits, but this is **highly unfair** to Bob. The rest of the talk provides a solution to this problem.

A concave utility function

- The previous example showed that the maximizing the cumulative rewards could result in an extremely unfair policy.
- To include fairness in the policy design, we maximize a strictly concave increasing function $\phi(\cdot)$ of the cumulative rewards $\boldsymbol{R}(t)$.
- The intuition is that, for a fixed cumulative rewards across all users, by Jensen's inequality, $\sum_i \phi(R_i)$ is maximized when $R_1 = R_2 = \ldots = R_m$.

The α-Fairness function

- The utility of any user with cumulative reward of R is given by the concave α-fair utility function:

$$\phi(R) = \phi_\alpha(R) \equiv \frac{R^{1-\alpha}}{1-\alpha}, \quad R \geq 0,$$

  for some constant $\alpha \geq 0$.

- The fairness parameter α induces a efficiency-fairness trade-off by incorporating a notion of *diminishing return* in the objective function.
  - $\alpha \to \infty$ corresponds to maxmin objective
  - $\alpha = 0$ corresponds to the unfair cumulative reward objective

The α-Fairness function

- The utility of any user with cumulative reward of R is given by the concave α-fair utility function:

$$\phi(R) = \phi_\alpha(R) \equiv \frac{R^{1-\alpha}}{1-\alpha}, \quad R \geq 0,$$

  for some constant $\alpha \geq 0$.

- The fairness parameter α induces a efficiency-fairness trade-off by incorporating a notion of *diminishing return* in the objective function.
  - $\alpha \to \infty$ corresponds to maxmin objective
  - $\alpha = 0$ corresponds to the unfair cumulative reward objective
- In the previous example, with $\alpha = 1/2$, under the optimal static policy, Alice gets a hit rate of 66.67% and Bob gets a hit rate of 16.67%.

A modified learning objective - c-Regret

- Suppose $\boldsymbol{y}^*$ is the optimal offline allocation. Then, our objective is to design a policy that achieves a sublinear bound for:

$$\mathrm{Regret}_T(c) \equiv \sum_{i=1}^{m} \phi(R_i^*(T)) - c \sum_{i=1}^{m} \phi(R_i(T)), \tag{2}$$

  for some small constant $c \geq 1$.

  The necessity for allowing $c > 1$ will be clear from the lower bound.

The difficulty of the problem

- The main difficulty stems from the non-separability of the objective across time.
  - The standard online convex optimization framework does not work
  - No online policy can achieve a sublinear regret. In fact, a stronger negative result holds

Theorem 1 (Lower Bound)

Any online policy with a sublinear c_α-regret must have

$$c_\alpha \geq \max_{0 \leq \eta \leq 1/2} \frac{\eta^{1-\alpha} + (1-\eta)^{1-\alpha}}{(1-\eta/2)^{1-\alpha} + (\eta/2)^{1-\alpha}} > 1, \quad 0 < \alpha < 1.$$

Designing the Online Fair Allocation (OFA) policy

- The first step is to linearize the objective using *policy-dependent* gradients.

- Using the concavity of the α-fairness function, we have the following bound to the $\beta^{1-\alpha}$-Regret

$$\text{Regret}_T(\beta^{1-\alpha}) \leq \beta^{-\alpha} \sum_{t=1}^{T} \left(\sum_i \langle \underbrace{\phi'(R_i(T))}_{\text{future-dependent!}} x_i(t), y_i^* - \beta y_i(t) \rangle \right).$$

  The parameter $\beta \geq 1$ will be fixed later.

The Online Fair Allocation (OFA) policy

- *Impossible* to know the terminal gradients $\phi'(R_i(T))$ for an online policy.

The Online Fair Allocation (OFA) policy

- *Impossible* to know the terminal gradients $\phi'(R_i(T))$ for an online policy.

- Solution: *Greedily* replace them by their causal counterpart $\phi'(R_i(t-1))$ and consider the following surrogate regret

$$\text{L-Regret}_T \equiv \sum_{t=1}^{T} \left(\sum_i \langle \phi'\big(R_i(t-1)\big) x_i(t), y_i^* - y_i(t) \rangle \right). \qquad (3)$$

- The above expression defines *some* legitimate online linear optimization problem

> Policy: Minimize *L*-Regret using AdaGrad

Connecting Regrets of the original problem to the surrogate problem

Lemma 2

For any horizon of length $T \geq 1$,

$$\mathrm{Regret}_T(c_\alpha) \leq (1-\alpha)^\alpha \cdot \text{L-Regret}_T + c_\alpha m, \tag{4}$$

where $c_\alpha \equiv (1-\alpha)^{-(1-\alpha)} < 1.445$.

- Proof involves splitting the regret expression Eq. (3) into the difference between the online and offline terms and then comparing each term with its corresponding causal counterpart.

OFA

Algorithm The Online Fair Allocation (`OFA`) Policy

Input: $0 \leq \alpha < 1$, $\{\boldsymbol{x}(t)\}_{t=1}^{T}$, Euclidean projection oracle $\Pi_{\Delta}(\cdot)$, an upper-bound D to the Euclidean diameter of the feasible set.

Output: Online resource allocation decisions $\{y_t\}_{t=1}^{T}$

$R_i \leftarrow 1, \forall i \in [m], S \leftarrow 0$ ▷ *Initialization*

for each round $t = 2 : T$: **do**

  $g_i \leftarrow \frac{x_i(t-1)}{R_i^{\alpha}}, \forall i \in [m]$ ▷ *Computing the gradient components for each agents*

  $g \leftarrow (g_1, g_2, \ldots, g_m)$ ▷ *Computing the full gradient*

  $S \leftarrow S + ||g||_2^2,$ ▷ *Accumulating the norm of the gradients*

  $y \leftarrow \Pi_{\Delta}\left(y + \frac{D}{2\sqrt{S}} g\right)$ ▷ *Updating the inclusion probabilities using OGA*

  [**Optional**] Sample a randomized integral allocation Y s.t. $\mathbb{E}[Y] = y$.

  The agents reveal their demand/ reward vectors $\{x_i(t)\}_{i \in [m]}$ for the current round.

  $R_i \leftarrow R_i + \langle x_i(t), y_i \rangle, \ \forall i \in [m].$ ▷ *Updating cumulative rewards*

end for each

Theorem 1

The OFA policy achieves the following regret bound for the surrogate problem:

$$\text{Regret}_T(1.445) \quad = \quad (1-\alpha)^{\alpha} \begin{cases} O(T^{1/2-\alpha}), & \text{if } \ 0 < \alpha < 1/2, \\ O(\sqrt{\log T}), & \text{if } \ \alpha = 1/2 \\ O(1), & \text{if } \ 1/2 < \alpha < 1. \end{cases}$$

Furthermore, under this policy, the cumulative rewards of each user increase linearly with time, *i.e.*, $R_i(T) = \Omega(T), \forall i, T$.

Adaptive regret bound for AdaGrad

Let $\Delta \subset \mathbb{R}^d$ be a closed and convex set with with diameter D. Consider a sequence of linear reward functions with gradients $\{\boldsymbol{g}_t\}_{t\geq 1}$. Run the Online Gradient Descent policy

$$\boldsymbol{x}_{t+1} = \Pi_{\Delta}(\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t)$$

with adaptive step sizes

$$\eta_t = \frac{D}{\sqrt{2\sum_{\tau=1}^{t} ||\boldsymbol{g}_\tau||_2^2}}, \ t \geq 1.$$

Then the regret can be upper-bounded as follows:

$$\text{Regret}_T \leq D\sqrt{2\sum_{t=1}^{T} ||\boldsymbol{g}_t||_2^2}. \tag{5}$$

Proof Sketch for Theorem 1

- Recall that the coefficients $\phi'(R_i(t-1))$ on round t depend on the past actions $\{\boldsymbol{y}(\tau)\}_{\tau=1}^{t}$ of the policy itself.
- The adaptive regret bound with AdaGrad scales with the norm of the gradients.
- We need to *simultaneously* control the regret *and* the norm of the gradients generated by the online policy.

Proof (contd.)

- The adaptive regret bound of AdaGrad yields

$$\text{L-Regret}_T = O\left(\sqrt{\sum_{t=1}^{T}\sum_{i}\phi'(R_i(t))^2}\right) = O\left(\sqrt{\sum_{t=1}^{T}\sum_{i}\frac{1}{R_i(t)^{2\alpha}}}\right). \quad (6)$$

A puzzle: The bound depends on $\{\boldsymbol{R}(t)\}_{t\geq 1}$, which are implicitly controlled by the policy. The policy is in turn determined by the sequence $\{\boldsymbol{R}(t)\}_{t\geq 1}$. How to bound the regret Eq. (6)?

Solution to the puzzle: Bootstrapping

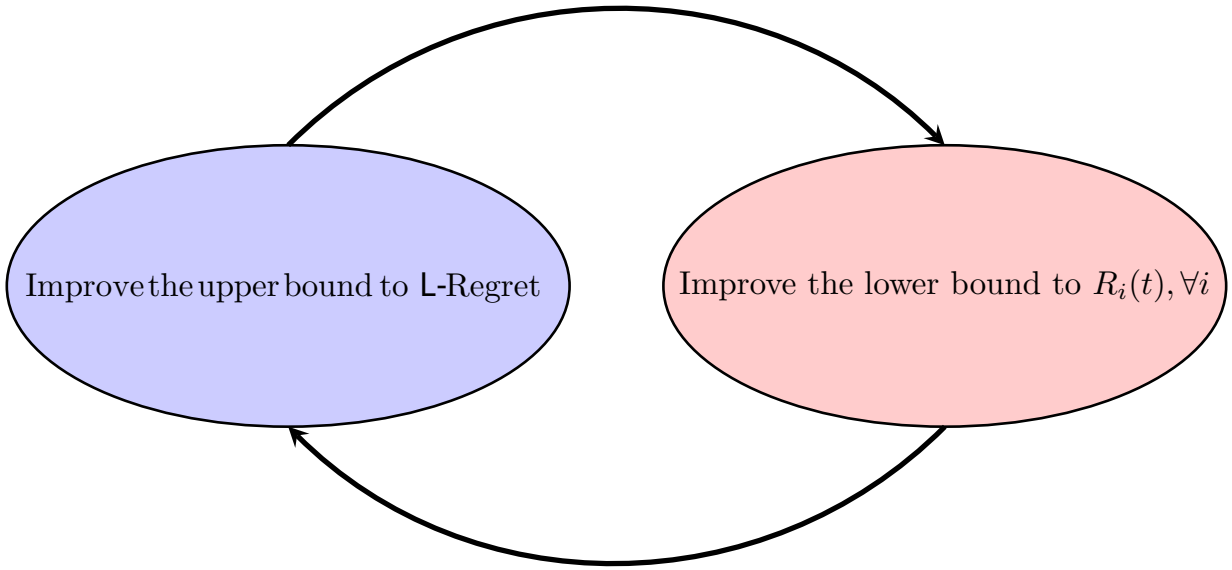


Figure: The *Bootstrapping* technique

Proof (contd.)

- Use the trivial bound $R_i(T) \geq 1, \forall i, \forall T$ to get a bound of $\text{L-Regret}_T = O(\sqrt{T})$.
- Using the upper bound on L-Regret_T, get the lower bound $R_i^{\alpha}(T) = \Omega(T^{\min(1/2,\alpha)}), \forall i \in [m]$.
- Plug this back into the upper bound for L-Regret_T to get the final regret bound.

Approximation Factor

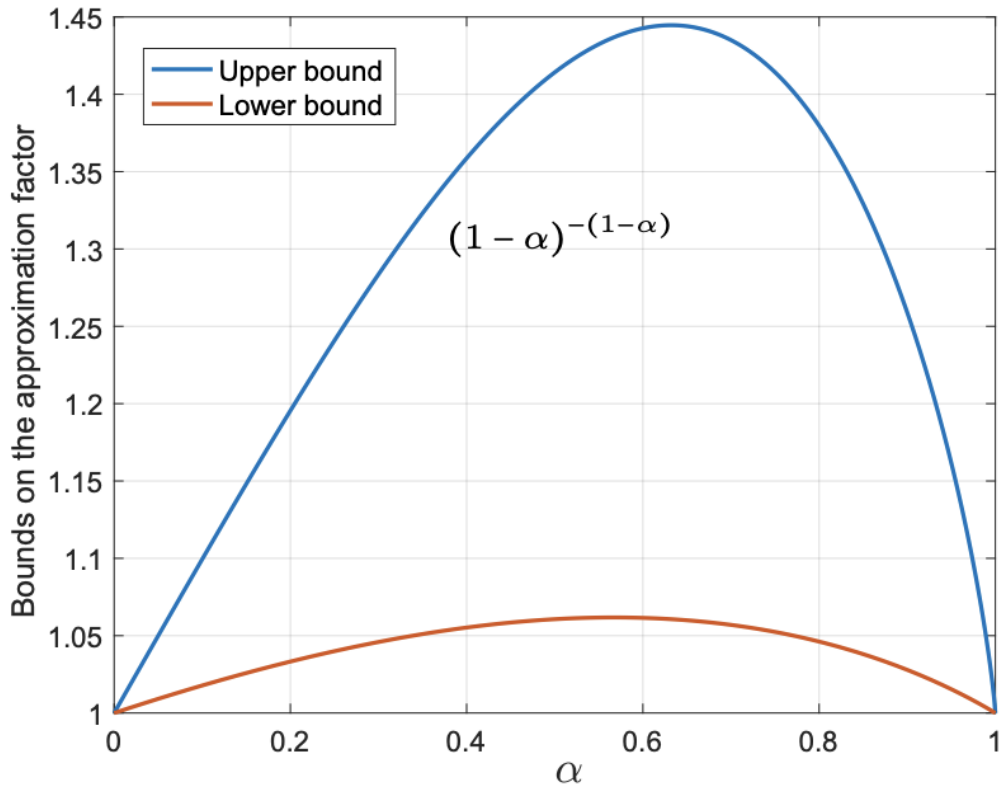


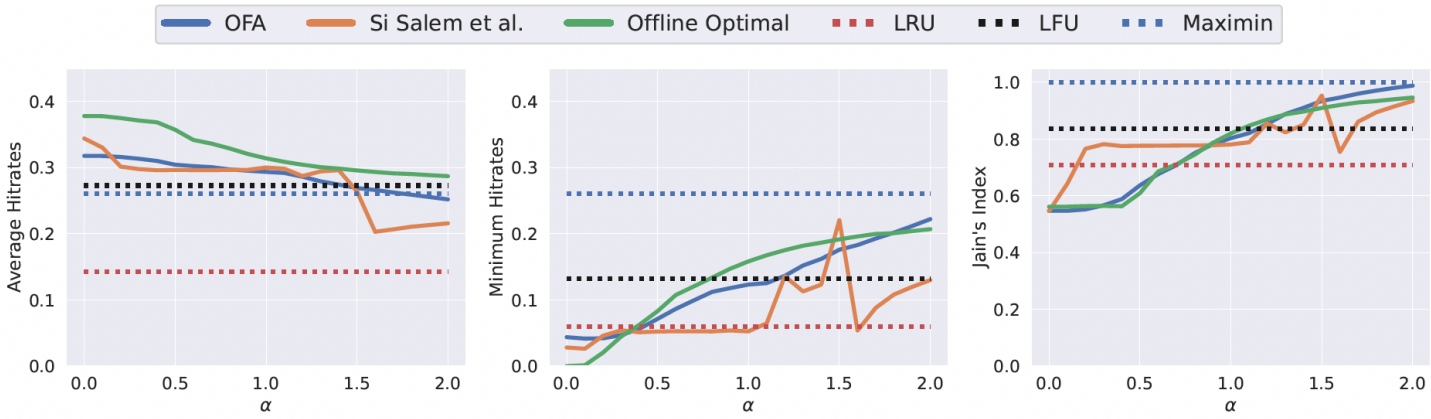
Figure: Comparison between the upper and lower bounds to the approximation factor as a function of $\alpha \in [0, 1]$

Datasets used in the experiments

1. CDN Traces (Berger, 2018): Horizon $T = 400$ rounds, number of users $m = 4$, cache size $C = 10$, and library size $N = 50$.

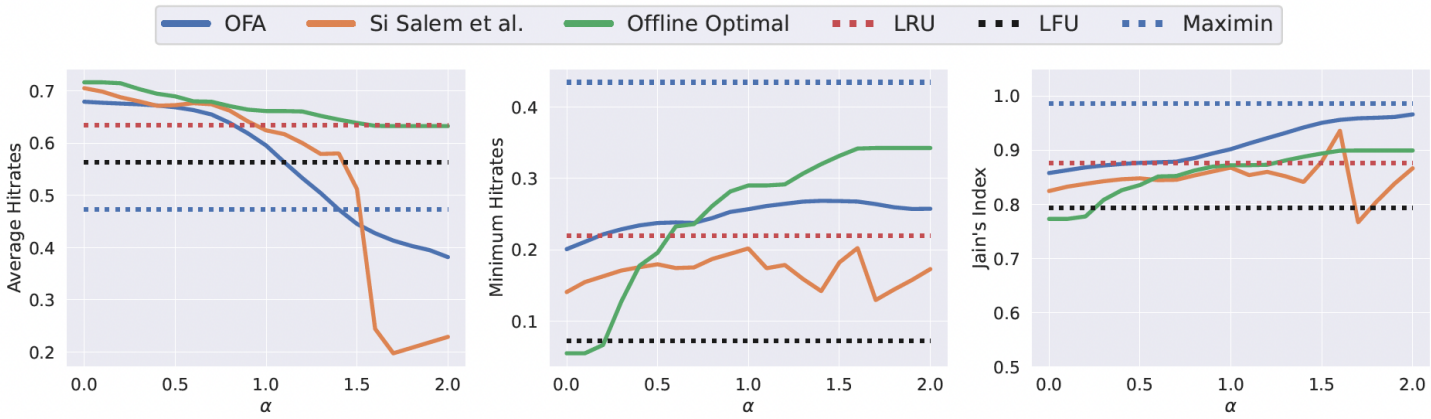
2. Synthetic Dataset:
Horizon $T = 1000$ rounds, number of users $m = 5$, cache size $C = 7$, and library size $N = 30$.

Experimental Results



(a) Average Hitrate (b) Minimum Hitrate (c) Jain's Index

Figure 2: Synthetic Dataset



(a) Average Hitrate (b) Minimum Hitrate (c) Jain's Index

Figure 3: CDN Dataset

References

[1] Sinha, Abhishek, Ativ Joshi, Rajarshi Bhattacharjee, Cameron Musco, and Mohammad Hajiesmaili. "No-regret Algorithms for Fair Resource Allocation." Advances in Neural Information Processing Systems 36 (2023).

[2] Abhishek Sinha. BanditQ - Fair Bandits with Guaranteed Rewards. In Uncertainty in Artificial Intelligence, Barcelona, Spain, PMLR, 2024.