# Online combinatorial linear optimization via a Frank-Wolfe based metarounding algorithm

Ryotaro Mitsuboshi[1,2] Kohei Hatano[1,2] Eiji Takimoto[1]

[1]Kyushu University

[2]RIKEN AIP

2023/11/11

# Online combinatorial optimization

The online linear optimization framework whose decision set $\mathcal{C} \subset \mathbb{R}_+^n$ is some combinatorial set.
For each round $t = 1, 2, \ldots, T$:


**Goal**

Minimize $\alpha$-regret $R_T(\alpha)$ with $\alpha \geq 1$.

$$R_T(\alpha) = \sum_{t=1}^{T} \boldsymbol{c}_t \cdot \boldsymbol{\ell}_t - \alpha \min_{\boldsymbol{c} \in \mathcal{C}} \sum_{t=1}^{T} \boldsymbol{c} \cdot \boldsymbol{\ell}_t$$

# Online combinatorial optimization

The online linear optimization framework whose decision set $\mathcal{C} \subset \mathbb{R}_+^n$ is some combinatorial set.
For each round $t = 1, 2, \ldots, T$:


**Goal**

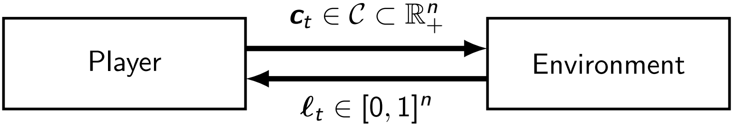
Minimize $\alpha$-regret $R_T(\alpha)$ with $\alpha \geq 1$.

$$R_T(\alpha) = \sum_{t=1}^{T} \boldsymbol{c}_t \cdot \ell_t - \alpha \min_{\boldsymbol{c} \in \mathcal{C}} \sum_{t=1}^{T} \boldsymbol{c} \cdot \ell_t$$

# Online combinatorial optimization

The online linear optimization framework whose decision set $\mathcal{C} \subset \mathbb{R}^n_+$ is some combinatorial set.

For each round $t = 1, 2, \ldots, T$:



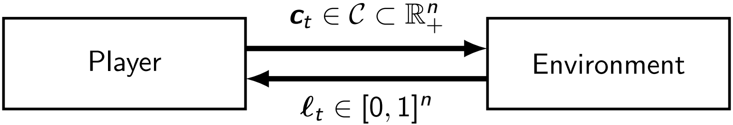

### Goal

Minimize $\alpha$-regret $R_T(\alpha)$ with $\alpha \geq 1$.

$$R_T(\alpha) = \sum_{t=1}^{T} \boldsymbol{c}_t \cdot \boldsymbol{\ell}_t - \alpha \min_{\boldsymbol{c} \in \mathcal{C}} \sum_{t=1}^{T} \boldsymbol{c} \cdot \boldsymbol{\ell}_t$$

# Online combinatorial optimization

The online linear optimization framework whose decision set $\mathcal{C} \subset \mathbb{R}_+^n$ is some combinatorial set.
For each round $t = 1, 2, \ldots, T$:




## Goal

Minimize $\alpha$-regret $R_T(\alpha)$ with $\alpha \geq 1$.

$$R_T(\alpha) = \sum_{t=1}^{T} \boldsymbol{c}_t \cdot \boldsymbol{\ell}_t - \alpha \min_{\boldsymbol{c} \in \mathcal{C}} \sum_{t=1}^{T} \boldsymbol{c} \cdot \boldsymbol{\ell}_t$$

# Comparison of the previous works and our contribution

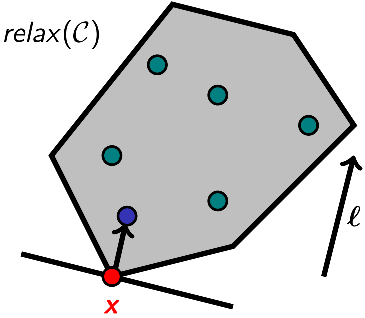
| Algorithm | Approx. oracle | # of oracle calls per trial | Regret |
|---|---|---|---|
| Kalai et al., '09 | Any | $O(T)$ | $O(T^{1/2})$ |
| Garber '17 | Any | $O(n^2 \ln T)$ | $O(T^{2/3})$ |
| Garber '17 | Any | $O(n^2 \sqrt{T} \ln T)$ | $O(T^{1/2})$ |
| Fujita et al., '13 | relax-based | $O(n^2 D_\infty^2 \ln(n)/\epsilon^2)$ | $O(T^{1/2})$ |
| **This work** | relax-based | $O(D_\infty^2 \ln(n)/\epsilon^2)$ | $O(T^{1/2})$ |

- $D_\infty = \max\{\|\boldsymbol{c}\|_\infty \mid \boldsymbol{c} \in \mathcal{C}\}$.
- Our work assumes the same setting as the one in Fujita et al.

# The main idea of Fujita et al., '13

**Idea.**

1. Run an online algorithm over *relax*$(\mathcal{C}) \supset \mathcal{C}$ to predict $\boldsymbol{x} \in$ *relax*$(\mathcal{C})$,
2. Project $\boldsymbol{x}$ back onto $\mathcal{C}$.




**Questions.**

- How to project onto $\mathcal{C}$?
- What condition is required to achieve low regret?

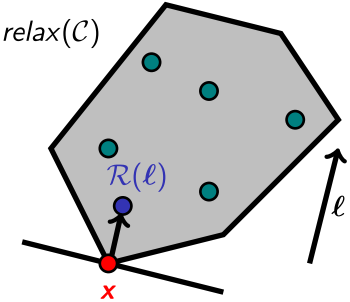
# A naïve idea: use an approx. oracle

For the projection, they assume the following oracle:

**Definition (Relax-based approximation Oracle)**

$\mathcal{R} : \mathbb{R}^n_+ \to \mathcal{C}$ is said to be a relax-based $\alpha$-approximation oracle for $(\mathit{relax}(\mathcal{C}), \mathcal{C})$ with $\mathit{relax}(\mathcal{C}) \supset \mathcal{C}$ if

$$\forall \boldsymbol{\ell} \in \mathbb{R}^n_+, \quad \mathcal{R}(\boldsymbol{\ell}) \cdot \boldsymbol{\ell} \leq \alpha \min_{\boldsymbol{x} \in \mathit{relax}(\mathcal{C})} \boldsymbol{x} \cdot \boldsymbol{\ell}.$$




1. Compute $\boldsymbol{x} \in \arg\min_{\boldsymbol{x} \in \mathit{relax}(\mathcal{C})} \boldsymbol{x} \cdot \boldsymbol{\ell}$.
2. Choose $\boldsymbol{c} \in \mathcal{C}$ such that $\boldsymbol{c} \cdot \boldsymbol{\ell} \leq \alpha \boldsymbol{x} \cdot \boldsymbol{\ell}$.

**Note:**
$\boldsymbol{\ell}$ is revealed after choosing $\boldsymbol{c} \in \mathcal{C}$, so one cannot use $\mathcal{R}$ directly.

# Metarounding

**Definition (Metarounding)**

Algorithm $\mathcal{M}$ is said to be an $\alpha$-metarounding for $(\mathcal{C}, \mathit{relax}(\mathcal{C}))$ if $\mathcal{M}$, when given $\boldsymbol{x} \in \mathit{relax}(\mathcal{C})$ as input, outputs a vector $\boldsymbol{c} \in \mathcal{C}$ randomly s.t.

$$\forall \boldsymbol{\ell} \in \mathbb{R}^n_+, \quad \mathbb{E}_{\mathcal{M}(\boldsymbol{x})}[\boldsymbol{c}] \cdot \boldsymbol{\ell} \leq \alpha \boldsymbol{x} \cdot \boldsymbol{\ell}.$$

The above is equivalent to $\mathbb{E}_{\mathcal{M}(\boldsymbol{x})}[\boldsymbol{c}] \leq \alpha \boldsymbol{x}$.

**Theorem (Carr et al., '02)**

*relax-based approx. oracle $\mathcal{R}$ exists $\implies$ metarounding $\mathcal{M}$ exists.*

# Metarounding

**Definition (Metarounding)**

Algorithm $\mathcal{M}$ is said to be an $\alpha$-metarounding for $(\mathcal{C}, \mathit{relax}(\mathcal{C}))$ if $\mathcal{M}$, when given $\boldsymbol{x} \in \mathit{relax}(\mathcal{C})$ as input, outputs a vector $\boldsymbol{c} \in \mathcal{C}$ randomly s.t.

$$\forall \boldsymbol{\ell} \in \mathbb{R}_+^n, \quad \mathbb{E}_{\mathcal{M}(\boldsymbol{x})}[\boldsymbol{c}] \cdot \boldsymbol{\ell} \leq \alpha \boldsymbol{x} \cdot \boldsymbol{\ell}.$$

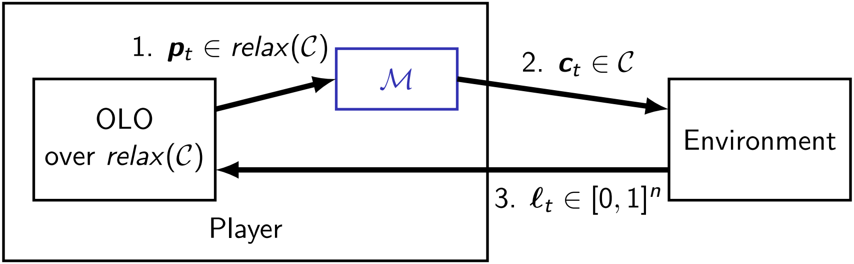
The above is equivalent to $\mathbb{E}_{\mathcal{M}(\boldsymbol{x})}[\boldsymbol{c}] \leq \alpha \boldsymbol{x}$.

**Theorem (Carr et al., '02)**

*relax-based approx. oracle $\mathcal{R}$ exists $\implies$ metarounding $\mathcal{M}$ exists.*

# Construct an online algorithm with Metarounding

If there exists an online algorithm for $relax(\mathcal{C})$, we can construct the one for $\mathcal{C}$ by fusing metarounding $\mathcal{M}$:




**Theorem (Fujita et al., '13)**

*Metarounding $\mathcal{M}$ for $(\mathcal{C}, relax(\mathcal{C}))$ exists*
$\Longrightarrow$ *the player above guarantees $R_T(\alpha) = O(\text{Regret bound for OLO})$.*

The above theorem implies $R_T(\alpha) = O(\sqrt{T})$.

# The idea in [Fujita et al., '13]

Estimate the approximation ratio $\alpha$ by formulating metarounding as an LP.

**Definition (Recap: Metarounding)**

$\mathcal{M}$ is an $\alpha$-metarounding for $(\mathcal{C}, \mathit{relax}(\mathcal{C}))$ if $\mathcal{M}$, when given $\boldsymbol{x} \in \mathit{relax}(\mathcal{C})$ as input, outputs a vector $\boldsymbol{c} \in \mathcal{C}$ randomly s.t.

$$\forall \boldsymbol{\ell} \in \mathbb{R}^n_+, \quad \mathbb{E}[\mathcal{M}(\boldsymbol{x})] \cdot \boldsymbol{\ell} \leq \alpha \boldsymbol{x} \cdot \boldsymbol{\ell}.$$

**Primal**

$$\min_{\beta, \boldsymbol{\lambda}} \beta \quad \text{s.t.} \quad \sum_{\boldsymbol{c} \in \mathcal{C}} \lambda_{\boldsymbol{c}} \boldsymbol{c} \leq \alpha \boldsymbol{x}, \quad \|\boldsymbol{\lambda}\|_1 = 1,\ \boldsymbol{\lambda} \geq \mathbf{0}.$$

**Dual**

$$\max_{\gamma, \boldsymbol{\ell}} \gamma \quad \text{s.t.} \quad \boldsymbol{\ell} \cdot \boldsymbol{c} \geq \gamma, \quad \forall \boldsymbol{c} \in \mathcal{C}, \quad \boldsymbol{\ell} \cdot \boldsymbol{x} = 1,\ \boldsymbol{\ell} \geq \mathbf{0}.$$

- They proposed an algorithm that solves the dual problem.
- The algorithm is designed based on SoftBoost [Warmuth et al., '07].
- The algorithm terminates in $O(n^2 D_\infty^2 \ln(n)/\epsilon^2)$ rounds.

# The idea in [Fujita et al., '13]

Estimate the approximation ratio $\alpha$ by formulating metarounding as an LP.

**Definition (Recap: Metarounding)**

$\mathcal{M}$ is an $\alpha$-metarounding for $(\mathcal{C}, \mathit{relax}(\mathcal{C}))$ if $\mathcal{M}$, when given $\boldsymbol{x} \in \mathit{relax}(\mathcal{C})$ as input, outputs a vector $\boldsymbol{c} \in \mathcal{C}$ randomly s.t.

$$\forall \boldsymbol{\ell} \in \mathbb{R}^n_+, \quad \mathbb{E}[\mathcal{M}(\boldsymbol{x})] \cdot \boldsymbol{\ell} \leq \alpha \boldsymbol{x} \cdot \boldsymbol{\ell}.$$

**Primal**

$$\min_{\beta, \boldsymbol{\lambda}} \beta \quad \text{s.t.} \quad \sum_{\boldsymbol{c} \in \mathcal{C}} \lambda_{\boldsymbol{c}} \boldsymbol{c} \leq \alpha \boldsymbol{x}, \quad \|\boldsymbol{\lambda}\|_1 = 1, \ \boldsymbol{\lambda} \geq \boldsymbol{0}.$$

**Dual**

$$\max_{\gamma, \boldsymbol{\ell}} \gamma \quad \text{s.t.} \quad \boldsymbol{\ell} \cdot \boldsymbol{c} \geq \gamma, \quad \forall \boldsymbol{c} \in \mathcal{C}, \quad \boldsymbol{\ell} \cdot \boldsymbol{x} = 1, \ \boldsymbol{\ell} \geq \boldsymbol{0}.$$

- They proposed an algorithm that solves the dual problem.
- The algorithm is designed based on SoftBoost [Warmuth et al., '07].
- The algorithm terminates in $O(n^2 D_\infty^2 \ln(n)/\epsilon^2)$ rounds.

# The idea in [Fujita et al., '13]

Estimate the approximation ratio $\alpha$ by formulating metarounding as an LP.

**Definition (Recap: Metarounding)**

$\mathcal{M}$ is an $\alpha$-metarounding for $(\mathcal{C}, \mathit{relax}(\mathcal{C}))$ if $\mathcal{M}$, when given $\boldsymbol{x} \in \mathit{relax}(\mathcal{C})$ as input, outputs a vector $\boldsymbol{c} \in \mathcal{C}$ randomly s.t.

$$\forall \boldsymbol{\ell} \in \mathbb{R}^n_+, \quad \mathbb{E}[\mathcal{M}(\boldsymbol{x})] \cdot \boldsymbol{\ell} \leq \alpha \boldsymbol{x} \cdot \boldsymbol{\ell}.$$

**Primal**

$$\min_{\beta, \boldsymbol{\lambda}} \beta \quad \text{s.t.} \quad \sum_{\boldsymbol{c} \in \mathcal{C}} \lambda_{\boldsymbol{c}} \boldsymbol{c} \leq \alpha \boldsymbol{x}, \quad \|\boldsymbol{\lambda}\|_1 = 1, \ \boldsymbol{\lambda} \geq \boldsymbol{0}.$$

**Dual**

$$\max_{\gamma, \boldsymbol{\ell}} \gamma \quad \text{s.t.} \quad \boldsymbol{\ell} \cdot \boldsymbol{c} \geq \gamma, \quad \forall \boldsymbol{c} \in \mathcal{C}, \quad \boldsymbol{\ell} \cdot \boldsymbol{x} = 1, \ \boldsymbol{\ell} \geq \boldsymbol{0}.$$

- They proposed an algorithm that solves the dual problem.
- The algorithm is designed based on SoftBoost [Warmuth et al., '07].
- The algorithm terminates in $O(n^2 D_\infty^2 \ln(n)/\epsilon^2)$ rounds.

# Our approach

Similar to ERLPBoost [Warmuth et al., '08], we add an entropy-like regularizer.

$$\max_{\gamma,\boldsymbol{\ell}} \gamma - \frac{1}{\eta}\sum_{i=1}^{n} x_i \ell_i \ln \frac{x_i \ell_i}{1/n} \quad \text{s.t.} \quad \boldsymbol{c} \cdot \boldsymbol{\ell} \geq \gamma, \quad \forall \boldsymbol{c} \in \mathcal{C},$$
$$\boldsymbol{x} \cdot \boldsymbol{\ell} = 1, \quad \boldsymbol{\ell} \geq \boldsymbol{0}.$$

Starting from $\mathcal{C}_0 = \emptyset$, for each round $k = 1, 2, \ldots, K$,

1. Solve the above problem over $\mathcal{C}_k$ to obtain the optimal solution $\ell_k$.
2. Update $\mathcal{C}_{k+1} = \mathcal{C}_k \cup \{\boldsymbol{c}_{k+1}\}$, where $\boldsymbol{c}_{k+1} = \mathcal{R}(\ell_k)$.

**Theorem (Convergence guarantee)**

*The above algo. finds an $\epsilon$-approx. solution in $O(D_\infty^2 \ln(n)/\epsilon^2)$ rounds.*

# Our approach

Similar to ERLPBoost [Warmuth et al., '08], we add an entropy-like regularizer.

$$\max_{\gamma,\boldsymbol{\ell}} \gamma - \frac{1}{\eta} \sum_{i=1}^{n} x_i \ell_i \ln \frac{x_i \ell_i}{1/n} \quad \text{s.t.} \quad \boldsymbol{c} \cdot \boldsymbol{\ell} \geq \gamma, \quad \forall \boldsymbol{c} \in \mathcal{C},$$
$$\boldsymbol{x} \cdot \boldsymbol{\ell} = 1, \quad \boldsymbol{\ell} \geq \mathbf{0}.$$

Starting from $\mathcal{C}_0 = \emptyset$, for each round $k = 1, 2, \ldots, K$,

1. Solve the above problem over $\mathcal{C}_k$ to obtain the optimal solution $\boldsymbol{\ell}_k$.
2. Update $\mathcal{C}_{k+1} = \mathcal{C}_k \cup \{\boldsymbol{c}_{k+1}\}$, where $\boldsymbol{c}_{k+1} = \mathcal{R}(\boldsymbol{\ell}_k)$.

**Theorem (Convergence guarantee)**

*The above algo. finds an $\epsilon$-approx. solution in $O(D_\infty^2 \ln(n)/\epsilon^2)$ rounds.*

# Our approach

Similar to ERLPBoost [Warmuth et al., '08], we add an entropy-like regularizer.

$$\max_{\gamma, \boldsymbol{\ell}} \gamma - \frac{1}{\eta} \sum_{i=1}^{n} x_i \ell_i \ln \frac{x_i \ell_i}{1/n} \quad \text{s.t.} \quad \boldsymbol{c} \cdot \boldsymbol{\ell} \geq \gamma, \quad \forall \boldsymbol{c} \in \mathcal{C}, \quad \boldsymbol{x} \cdot \boldsymbol{\ell} = 1, \quad \boldsymbol{\ell} \geq \mathbf{0}.$$

Starting from $\mathcal{C}_0 = \emptyset$, for each round $k = 1, 2, \ldots, K$,

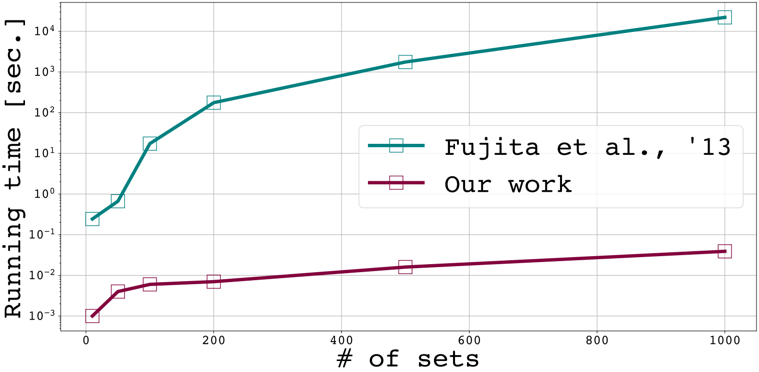
1. Solve the above problem over $\mathcal{C}_k$ to obtain the optimal solution $\boldsymbol{\ell}_k$.
2. Update $\mathcal{C}_{k+1} = \mathcal{C}_k \cup \{\boldsymbol{c}_{k+1}\}$, where $\boldsymbol{c}_{k+1} = \mathcal{R}(\boldsymbol{\ell}_k)$.

**Theorem (Convergence guarantee)**

*The above algo. finds an $\epsilon$-approx. solution in $O(D_\infty^2 \ln(n)/\epsilon^2)$ rounds.*

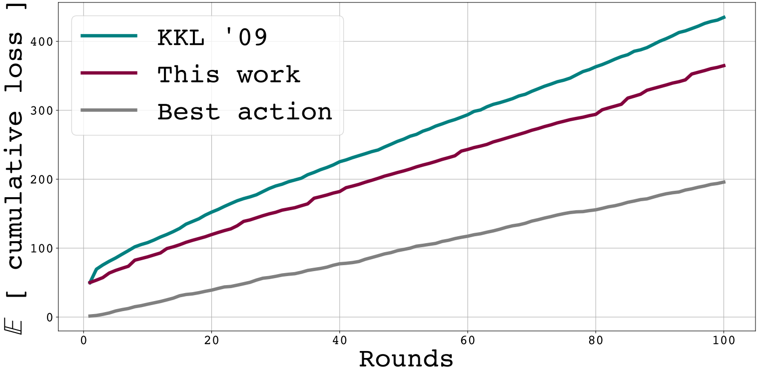
# Experiment

- Randomly generated set-cover instance.
- $n = 20$, varying the set sizes over $\{200, 400, 600, 800, 1000\}$.



# Experiment

- Randomly generated set-cover instance.
- $n = 20$ and $|\mathcal{C}| = 100$.



# Conclusion

- We proposed a metarounding algorithm which converges with rate $O(D_\infty^2 \ln(n)/\epsilon^2)$.
- Our algorithm is faster than the previous one under the same setting.

**Questions.**

- Does the proposed approach applicable in Bandit setting?
- Is the iteration bound optimal?