ONLINE LEARNING FOR PANDORA'S BOX AND MIN SUM SET COVER

Evangelia Gergatsouli, Christos Tzamos

University of Wisconsin-Madison

ICML, July 2022

A SEARCH PROBLEM

Find the best alternative!

A SEARCH PROBLEM

Find the best alternative!

- Information is not free
- Explore alternatives (open boxes)
- Stop anytime and take best so far

} **Strategy**

↓

Partially adaptive: fixed order, arbitrary stopping rule

A SEARCH PROBLEM

Find the best alternative!

42min ?? 50min 35min ??

Opening cost: 3
Final option: Route 4
Total cost: 32+3

- Information is not free
- Explore alternatives (open boxes)
- Stop anytime and take best so far

} **Strategy**

↓

Partially adaptive: fixed order, arbitrary stopping rule

A SEARCH PROBLEM

Find the best alternative!

42min | ?? | 50min | 35min | ??

Opening cost: 3
Final option: Route 4
Total cost: 32+3

- Information is not free
- Explore alternatives (open boxes)
- Stop anytime and take best so far

} **Strategy**

↓

Partially adaptive: fixed order, arbitrary stopping rule

Goal:
Find minimum cost strategy

PREVIOUS WORK

Stochastic version

- Independent boxes:
  - Choose 1 box: optimal strategy [Weitz 1979]
  - Choose k/matroid: optimal strategy [Singla 2018]
- Correlated boxes: (vs PA)[Chawla et al. 2020]
  - Choose 1 box: 9.22-approx
  - Choose k/matroid: O(1)/O(log k)

Online version

- Special case (Min Sum Set Cover)
  - Choose 1: 11.73-approx [Fotakis et al 2020]
  - Choose 1: log n [Munagala et al 2005]

PREVIOUS WORK

Stochastic version

- Independent boxes:
  - Choose 1 box: optimal strategy [Weitz 1979]
  - Choose k/matroid: optimal strategy [Singla 2018]
- Correlated boxes: (vs PA)[Chawla et al. 2020]
  - Choose 1 box: 9.22-approx
  - Choose k/matroid: O(1)/O(log k)

Match results in online version

Online version

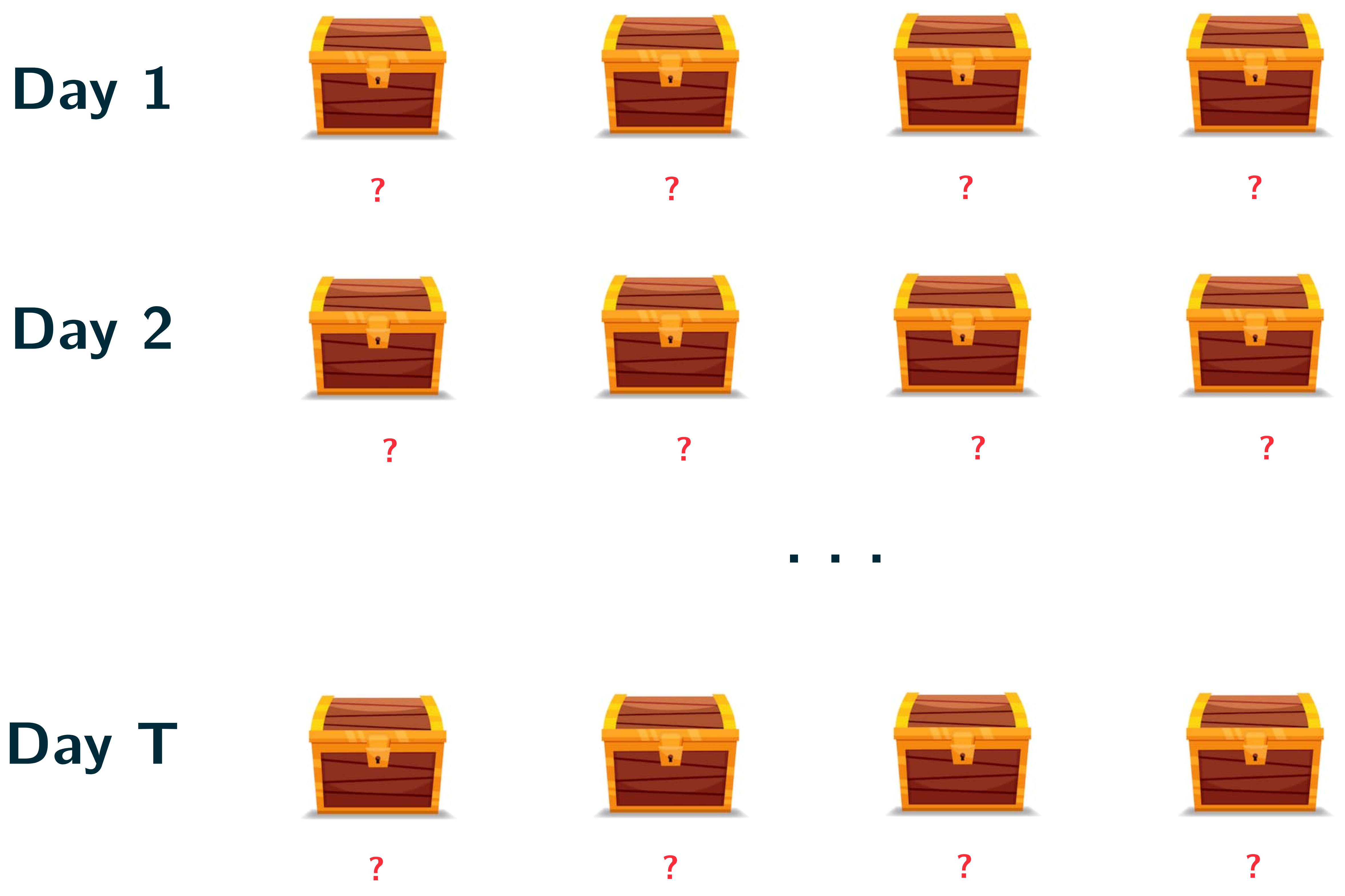
- Special case (Min Sum Set Cover)
  - Choose 1: 11.73-approx [Fotakis et al 2020]
  - Choose 1: log n [Munagala et al 2005]

Improve results + Simpler algorithm

This work: Online Pandora's Box over T time steps

AN ONLINE SEARCH PROBLEM

Online Pandora's Box over T time steps?



For each day t:

1. New realization of values in boxes
2. Pick a strategy $\mathscr{A}^t$
3. Play x_t according to $\mathscr{A}^t$
4. Receive loss $f^t(x_t)$
5. See loss function f^t on all x's

Goal:

Obtain α-approximate no regret algorithm vs hindsight optimal

AN ONLINE SEARCH PROBLEM

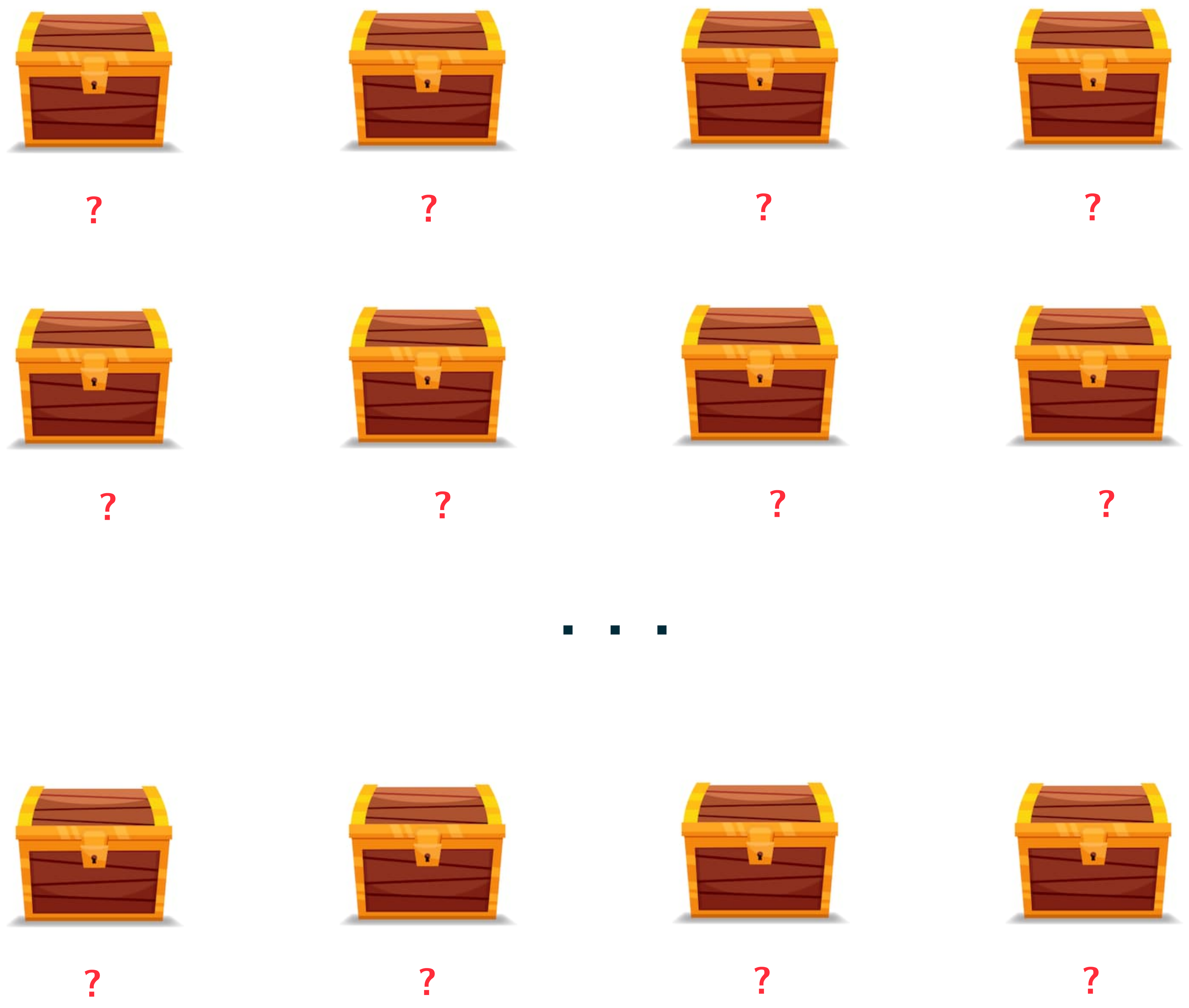
Online Pandora's Box over T time steps?

Day 1

Day 2

. . .

Day T



For each day t:

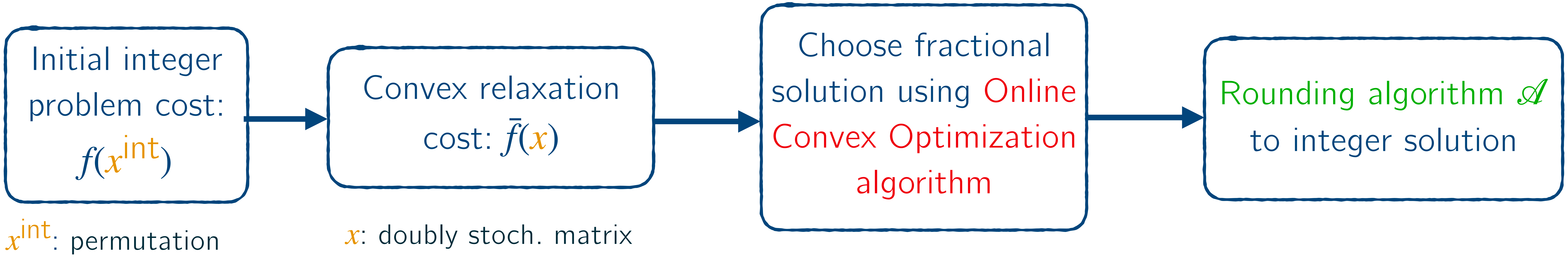
1. New realization of values in boxes
2. Pick a strategy $\mathscr{A}^t$
3. Play x_t according to $\mathscr{A}^t$
4. Receive loss $f^t(x_t)$
5. See loss function f^t on all x's

Goal:

Obtain α-approximate no regret algorithm vs hindsight optimal

$$\left\{ \frac{1}{T} \sum_{t \in [T]} \mathscr{A}(t) - \alpha \mathrm{OPT}(t) \right\} \to 0$$

OUR FRAMEWORK (1/2)



Main algorithm:

- $f^t(x)$ is the fractional objective function
- For each round $t \in [T]$ do
  - Set $x_t = \text{OCO}(f^1, \ldots, f^{t-1})$
  - Round x_t to x_t^{int} according to algorithm $\mathscr{A}$
  - Receive loss $f^t(x_t^{\text{int}})$

OUR FRAMEWORK (1/2)

Initial integer problem cost: $f(x^{\text{int}})$ → Convex relaxation cost: $\bar{f}(x)$ → Choose fractional solution using Online Convex Optimization algorithm → Rounding algorithm $\mathscr{A}$ to integer solution

x^{int}: permutation

x: doubly stoch. matrix

Main algorithm:

- $f^t(x)$ is the fractional objective function
- For each round $t \in [T]$ do
  - Set $x_t = \text{OCO}(f^1, \ldots, f^{t-1})$
  - Round x_t to x_t^{int} according to algorithm $\mathscr{A}$
  - Receive loss $f^t(x_t^{\text{int}})$

Regret guarantee $R(T)$

+

α-approx rounding algorithm

Theorem 3.1

$\alpha R(T)$ approx. regret

OUR FRAMEWORK (1/2)

Initial integer problem cost: $f(x^{\text{int}})$ → Convex relaxation cost: $\bar{f}(x)$ → Choose fractional solution using Online Convex Optimization algorithm → Rounding algorithm $\mathscr{A}$ to integer solution

x^{int}: permutation

x: doubly stoch. matrix

Main algorithm:

- $f^t(x)$ is the fractional objective function
- For each round $t \in [T]$ do
  - Set $x_t = \text{OCO}(f^1, \ldots, f^{t-1})$
  - Round x_t to x_t^{int} according to algorithm $\mathscr{A}$
  - Receive loss $f^t(x_t^{\text{int}})$

Regret guarantee $R(T)$

\+

α-approx rounding algorithm

Theorem 3.1

$\alpha R(T)$ approx. regret

Optimize the two components independently!

OUR FRAMEWORK (2/2)

Regret guarantee $R(T)$

- Use **F**ollow **T**he **R**egularized **L**eader family
- Set $OCO = \min_x \sum_{\tau=1}^{t-1} f^\tau(x) + \text{Regularizer}(x)$
- Choose regularizer $= \sum_{i,t} x_{it} \log x_{it}$

Theorem 3.3
OCO Algorithm is no regret

α-approximation algorithm

- Use randomized rounding [Chawla et al. 2020]
- Guarantee $\mathbb{E}[\bar{f}(x)] \leq \alpha f(x^{\text{int}})$
- Does not depend on f^t

Corollary C.0.1
There exists a 9.22-approximation rounding algorithm for selecting 1 box

OUR FRAMEWORK (2/2)

Regret guarantee $R(T)$

- Use **F**ollow **T**he **R**egularized **L**eader family
- Set $OCO = \min_x \sum_{\tau=1}^{t-1} f^\tau(x) + \text{Regularizer}(x)$
- Choose regularizer $= \sum_{i,t} x_{it} \log x_{it}$

α-approximation algorithm

- Use randomized rounding [Chawla et al. 2020]
- Guarantee $\mathbb{E}[\bar{f}(x)] \leq \alpha f(x^{\text{int}})$
- Does not depend on f^t

Corollary 3.3.1

Algorithm is 9.22-approx no regret

RESULTS AND EXTENSIONS

- More involved constraints:
  - Choose 1 box
  - Choose k boxes
  - Choose a matroid basis of size k

	1 box	**k boxes**	**Matroid basis, size k**
α-approx. Regret	$\alpha = 9.22$	$\alpha = O(1)$	$\alpha = O(\log k)$

RESULTS AND EXTENSIONS

- More involved constraints:
  - Choose 1 box
  - Choose k boxes
  - Choose a matroid basis of size k

	1 box	**k boxes**	**Matroid basis, size k**
α-approx. Regret	$\alpha = 9.22$	$\alpha = O(1)$	$\alpha = O(\log k)$

For each day t:

1. New realization of values in boxes
2. Pick a strategy $\mathscr{A}^t$
3. Play x_t according to $\mathscr{A}^t$
4. Receive loss $f^t(x_t)$
5. See loss function f^t on all x's

What if not full information?

BANDIT SETTING

1. New realization of values in boxes
2. Pick a strategy $\mathscr{A}^t$
3. Play x_t according to $\mathscr{A}^t$
4. Receive loss $f^t(x_t)$

5. See loss function on all x_t

Full information setting

5. See loss function **only** on x_t

Bandit setting

RESULTS

Same results for full info & bandit !

How? Balance explore/FTRL steps

RESULTS

Same results for full info & bandit !

How? Balance explore/FTRL steps

Main algorithm:

- Split $[T]$ into intervals $\mathcal{I}_i$, choose uniformly random $t_p \in [\mathcal{I}_i]$, $\mathcal{R} = \emptyset$
- For each interval $\mathcal{I}_i$ and each time $t \in \mathcal{I}_i$
  - If $t = t_p$
    - Open all boxes, include t_p in $\mathcal{R}$
  - Else
    - Set $x_t = \min_x \sum_{\tau \in \mathcal{R}} f^\tau(x) + \text{Regularizer}(x)$
    - Round x_t to x_t^{int} according to algorithm $\mathcal{A}$

RESULTS

Same results for full info & bandit !

How? Balance explore/FTRL steps

Main algorithm:

- Split $[T]$ into intervals $\mathcal{I}_i$, choose uniformly random $t_p \in [\mathcal{I}_i]$, $\mathcal{R} = \emptyset$
- For each interval $\mathcal{I}_i$ and each time $t \in \mathcal{I}_i$
  - If $t = t_p$
    - Open all boxes, include t_p in $\mathcal{R}$
  - Else
    - Set $x_t = \min_x \sum_{\tau \in \mathcal{R}} f^\tau(x) + \text{Regularizer}(x)$
    - Round x_t to x_t^{int} according to algorithm $\mathcal{A}$

Theorem 4.1

In the bandit setting, OCO Algorithm is no regret

CONCLUSION

	1 box	k boxes	Matroid basis, size k
α-approx. Regret	$\alpha = 9.22$	$\alpha = O(1)$	$\alpha = O(\log k)$
α-approx. Regret	$\alpha = 3.16$	$\alpha = 12.64$	$\alpha = O(\log k)$

CONCLUSION

Full information & bandit

Against PA

	1 box	k boxes	Matroid basis, size k
α-approx. Regret	$\alpha = 9.22$	$\alpha = O(1)$	$\alpha = O(\log k)$
α-approx. Regret	$\alpha = 3.16$	$\alpha = 12.64$	$\alpha = O(\log k)$

CONCLUSION

Full information & bandit

		1 box	k boxes	Matroid basis, size k
Against PA	**α-approx. Regret**	$\alpha = 9.22$	$\alpha = O(1)$	$\alpha = O(\log k)$
Against NA	**α-approx. Regret**	$\alpha = 3.16$	$\alpha = 12.64$	$\alpha = O(\log k)$

CONCLUSION

	1 box	k boxes	Matroid basis, size k
Full information & bandit Against PA — α-approx. Regret	$\alpha = 9.22$	$\alpha = O(1)$	$\alpha = O(\log k)$
Against NA — α-approx. Regret	$\alpha = 3.16$	$\alpha = 12.64$	$\alpha = O(\log k)$

Different ellipsoid-based algorithm

CONCLUSION

Full information & bandit

		1 box	k boxes	Matroid basis, size k
Against PA	**α-approx. Regret**	$\alpha = 9.22$	$\alpha = O(1)$	$\alpha = O(\log k)$
Against NA	**α-approx. Regret**	$\alpha = 3.16$	$\alpha = 12.64$	$\alpha = O(\log k)$

Different ellipsoid-based algorithm

Thank you!